源代码

package org.icscn.server.standalone.net.dao;

import java.sql.ResultSet;

import java.util.ArrayList;

import java.util.Iterator;

import java.util.List;

import org.icscn.server.standalone.db.DaoUtil;

import org.icscn.server.standalone.net.model.GetPrivRepMessage;

import org.icscn.server.standalone.net.model.GetSecRepMessage;

import org.icscn.server.standalone.net.model.GetUserRepMessage;

import org.icscn.server.standalone.net.model.SetSecReqMessage;

import org.icscn.server.standalone.net.model.SetUserReqMessage;

import com.aiyc.framework.utils.StringUtils;

public class UserDao {

/\*\*

\* 获取头像文件

\*/

public String getUserHead(String userNo) throws Exception {

String sql = "select USER\_IMG from k\_user\_ext where USER\_NO=? ";

ResultSet rs = DaoUtil.queryData(sql, new Object[] { userNo });

if (rs.next()) {

return rs.getString("USER\_IMG");

}

return "";

}

/\*\*

\* 修改头像

\*

\* @param model

\* @return

\* @throws Exception

\*/

public void modifyUserHead(String userNo, String head) throws Exception {

String sql = "select 1 from k\_user\_ext where USER\_NO=? ";

ResultSet rs = DaoUtil.queryData(sql, new Object[] { userNo });

if (rs.next()) {// //更新

sql = "insert into arc\_k\_user\_ext (user\_no,user\_img,up\_date,arc\_date)"

+ " select user\_no,user\_img,up\_date ,now() from k\_user\_ext where user\_no=? ";

DaoUtil.executeQuery(sql, new Object[] { userNo });

sql = " update k\_user\_ext set USER\_IMG=?,UP\_DATE=now() where user\_no=? ";

DaoUtil.executeQuery(sql, new Object[] { head, userNo });

} else {// /记录

sql = "insert into k\_user\_ext(user\_no,user\_img,up\_date) values(?,?,now())";

DaoUtil.executeQuery(sql, new Object[] { userNo, head });

}

}

package org.icscn.server.standalone.net.dao;

import java.sql.ResultSet;

import org.icscn.server.standalone.db.DaoUtil;

public class LoginDao {

public ResultSet queryUser(String userName, String userPass)

throws Exception {

String sql = "select user\_id from k\_user where user\_no=? and user\_pass=? ";

return DaoUtil.queryData(sql, new Object[] { userName, userPass });

}

public ResultSet queryUserByName(String userName) throws Exception {

String sql = "select user\_id from k\_user where user\_no=? ";

return DaoUtil.queryData(sql, new Object[] { userName });

}

public ResultSet queryUserByOpenID(String openId) throws Exception {

String sql = "select user\_id from k\_user k where exists (select 1 from "

+ " k\_user\_rela r where r.user\_id=k.user\_id and r.open\_id=? ) ";

return DaoUtil.queryData(sql, new Object[] { openId });

}

}

package org.icscn.server.standalone.net.dao;

import org.icscn.server.standalone.db.DaoUtil;

import org.icscn.server.standalone.net.model.RegReqMessage;

import com.aiyc.framework.utils.StringUtils;

public class RegDao {

public int addUser(RegReqMessage req) throws Exception {

String sql = "INSERT INTO k\_user\n"

+ "\t\t(user\_no, user\_name, user\_pass, base\_org, login\_ip, status, is\_admin, css,\n"

+ "\t\t phone, mail,reg\_date,sex,act\_Name,open\_id)\n" + "VALUES\n"

+ "\t\t(?, ?, ?, ?, '', '01', '02', 'blue', ?, ?,now(),?,?,?)";

String userName="";

if(StringUtils.isNullOrBank(req.MESSAGE.USER\_NAME)){

userName= req.MESSAGE.USER\_NO;

}

else{

userName= req.MESSAGE.USER\_NAME;

}

return DaoUtil.executeQuery(sql, new Object[] {

req.MESSAGE.USER\_NO, userName, req.MESSAGE.USER\_PWD, "",

"1".equals(req.MESSAGE.IS\_PHONE) ? req.MESSAGE.USER\_NO : "",

"1".equals(req.MESSAGE.IS\_MAIL) ? req.MESSAGE.USER\_NO : "",

req.MESSAGE.SEX, req.MESSAGE.ACT\_NAME,

req.MESSAGE.OPEN\_ID

});

}

}

package org.icscn.server.standalone.net.dao.area;

import java.sql.ResultSet;

import java.util.ArrayList;

import java.util.List;

import org.icscn.server.standalone.db.DaoUtil;

import org.icscn.server.standalone.net.model.area.SetAreaReqMessage;

import org.icscn.server.standalone.net.model.area.SetBuildReqMessage;

import org.icscn.server.standalone.net.model.area.SetDoorReqMessage;

import org.icscn.server.standalone.net.model.area.SetFloorReqMessage;

import org.icscn.server.standalone.net.model.area.SetGraphReqMessage;

import com.aiyc.framework.utils.StringUtils;

public class GraphDao {

/\*\*

\* 通过地图查询整个区域

\*/

public ResultSet queryArea(String mapNo) throws Exception {

String sql = " select `ID`, " + "`FLOOR\_NO` , " + "`BUILD\_NO` , "

+ "`AREA\_NO`, " + "`IMG\_URL`, " + "`DOORS` , "

+ "`MAP\_NO` from d\_floor where map\_no=?";

return DaoUtil.queryData(sql, new Object[] { mapNo });

}

/\*\*

\* 保存图形

\*/

public void saveGraph(SetGraphReqMessage model) throws Exception {

String sql = "insert into map ( mapId , mapName ,mapURL ,"

+ "mapx0,mapy0,mapx1,mapy1,centerx,centery,maplevel,minlevel,maxlevel )"

+ "" + "values(" + DaoUtil.makePix(12) + ")";

String[] bounds = model.MESSAGE.MAP\_BOUNDS.split(",");

String[] ceters = model.MESSAGE.MAP\_CENTER.split(",");

DaoUtil.executeQuery(sql, new Object[] { model.MESSAGE.MAP\_NO,

model.MESSAGE.MAP\_NAME, model.MESSAGE.MAP\_URL, bounds[0],

bounds[1], bounds[2], bounds[3], ceters[0], ceters[1],

model.MESSAGE.MAP\_LEVEL, model.MESSAGE.MIN\_LEVEL,

model.MESSAGE.MAX\_LEVEL });

}

/\*\*

\* 查询seller

\*/

public boolean isSeller(String areaNo, String buildNo, String floorNo,

String doorNo) throws Exception {

String sql = "select 1 from s\_seller r where `AREA\_NO`=? ";

List list = new ArrayList();

list.add(areaNo);

if (!StringUtils.isNullOrBank(buildNo)) {

sql += " and `BUILD\_NO`=? ";

list.add(buildNo);

}

if (!StringUtils.isNullOrBank(floorNo)) {

sql += " and `FLOOR\_NO`=? ";

list.add(floorNo);

}

if (!StringUtils.isNullOrBank(doorNo)) {

sql += " and `DOOR\_NO`=? ";

list.add(doorNo);

}

return DaoUtil.queryData(sql, list.toArray()).next();

}

/\*\*

\* 查询地图是否还在使用

\*/

public boolean ismapUse(String mapNo) throws Exception {

String sql = "select 1 from d\_floor r where r.map\_no=? ";

return DaoUtil.queryData(sql, new Object[] { mapNo }).next();

}

/\*\*

\* 删除图形

\*/

public void delGraph(String mapNo) throws Exception {

String sql = "delete from map where mapid=? ";

DaoUtil.executeQuery(sql, new Object[] { mapNo });

}

/\*\*

\* 删除区域

\*/

public void delArea(String areaNo, boolean sub) throws Exception {

String sql = "delete from d\_area where AREA\_NO=? ";

DaoUtil.executeQuery(sql, new Object[] { areaNo });

if (sub) {

sql = "delete from d\_building where AREA\_NO=? ";

DaoUtil.executeQuery(sql, new Object[] { areaNo });

sql = "delete from d\_floor where AREA\_NO=? ";

DaoUtil.executeQuery(sql, new Object[] { areaNo });

sql = "delete from d\_door where AREA\_NO=? ";

DaoUtil.executeQuery(sql, new Object[] { areaNo });

}

}

/\*\*

\* 删除楼宇

\*/

public void delBuild(String areaNo, String buildNo, boolean sub)

throws Exception {

String sql = "delete from d\_building where AREA\_NO=? and BUILDING\_NO=? ";

DaoUtil.executeQuery(sql, new Object[] { areaNo, buildNo });

if (sub) {

sql = "delete from d\_floor where AREA\_NO=? and BUILDING\_NO=? ";

DaoUtil.executeQuery(sql, new Object[] { areaNo, buildNo });

sql = "delete from d\_door where AREA\_NO=? and BUILDING\_NO=? ";

DaoUtil.executeQuery(sql, new Object[] { areaNo, buildNo });

}

}

/\*\*

\* 删除楼层

\*/

public void delFloor(String areaNo, String buildNo, String floorNo,

boolean sub) throws Exception {

String sql = "delete from d\_floor where AREA\_NO=? and BUILDING\_NO=? and FLOOR\_NO=? ";

DaoUtil.executeQuery(sql, new Object[] { areaNo, buildNo, floorNo });

if (sub) {

sql = "delete from d\_door where AREA\_NO=? and BUILDING\_NO=? and FLOOR\_NO=? ";

DaoUtil

.executeQuery(sql,

new Object[] { areaNo, buildNo, floorNo });

}

}

/\*\*

\* 删除房间

\*/

public void delDoor(String areaNo, String buildNo, String floorNo,

String doorNo) throws Exception {

String sql = "delete from d\_door where AREA\_NO=? and BUILDING\_NO=? and FLOOR\_NO=? and doorNo=? ";

DaoUtil.executeQuery(sql, new Object[] { areaNo, buildNo, floorNo,

doorNo });

}

/\*\*

\* 保存区域信息

\*/

public void saveAreaInfo(SetAreaReqMessage model) throws Exception {

String sql = " insert into d\_area( `AREA\_NO` , `AREA\_NAME` ,"

+ "`AREA\_ADDR` , `BUILD\_NUM` , `AREA\_TYPE` ,"

+ "`P\_AREA\_NO` , `IMG\_URL`) values(" + DaoUtil.makePix(7) + ")";

DaoUtil.executeQuery(sql, new Object[] { model.MESSAGE.AREA\_NO,

model.MESSAGE.AREA\_NAME, model.MESSAGE.AREA\_ADDR,

model.MESSAGE.BUILD\_NUM, model.MESSAGE.AREA\_TYPE,

model.MESSAGE.P\_AREA\_NO, model.MESSAGE.IMG\_URL,

});

}

/\*\*

\* 保存楼宇

\*/

public void saveBuildInfo(SetBuildReqMessage model) throws Exception {

String sql = " insert into `d\_building`( `BUILDING\_NO` ,"

+ "`AREA\_NO` ," + "`BUILDING\_NAME` ," + "`BUILDING\_DESC` ,"

+ "`FLOORS` ," + "`FLOOR\_PIX` ," + "`JD` ," + " `WD` ,"

+ "`IMG\_URL`) values(" + DaoUtil.makePix(9) + ")";

DaoUtil.executeQuery(sql, new Object[] { model.MESSAGE.BUILD\_NO,

model.MESSAGE.AREA\_NO, model.MESSAGE.BUILDING\_NAME,

model.MESSAGE.BUILDING\_DESC, model.MESSAGE.FLOORS,

model.MESSAGE.FLOOR\_PIX, model.MESSAGE.JD, model.MESSAGE.WD,

model.MESSAGE.IMG\_URL

});

}

/\*\*

\* 保存楼层

\*/

public void saveFloorInfo(SetFloorReqMessage model) throws Exception {

String sql = " insert into `d\_floor`( `FLOOR\_NO` , " + "`BUILD\_NO` , "

+ "`AREA\_NO`, " + "`IMG\_URL`, " + "`DOORS` , "

+ "`MAP\_NO` ) values(" + DaoUtil.makePix(6) + ")";

DaoUtil.executeQuery(sql, new Object[] { model.MESSAGE.FLOOR\_NO,

model.MESSAGE.BUILD\_NO, model.MESSAGE.AREA\_NO, model.MESSAGE.IMG\_URL,

model.MESSAGE.DOORS, model.MESSAGE.MAP\_NO });

}

/\*\*

\* 保存房间

\*/

public void saveDoorInfo(SetDoorReqMessage model) throws Exception {

String sql = " insert into `d\_door`( `DOOR\_NO` , " + "`FLOOR\_NO` , "

+ "`BUILD\_NO` , " + "`AREA\_NO` ) values(" + DaoUtil.makePix(4)

+ ")";

DaoUtil.executeQuery(sql, new Object[] { model.MESSAGE.DOOR\_NO,

model.MESSAGE.FLOOR\_NO, model.MESSAGE.BUILD\_NO, model.MESSAGE.AREA\_NO,

});

}

/\*\*

\*

\* @param mapNo

\* @return

\* @throws Exception

\*/

public ResultSet getGraph(String mapNo) throws Exception {

String sql = "select mapId MAP\_NO, mapName MAP\_NAME ,mapURL MAP\_URL,"

+ "mapx0,mapy0,mapx1,mapy1,centerx,centery,maplevel,minlevel,maxlevel from map where mapId=?";

return DaoUtil.queryData(sql, new Object[] { mapNo });

}

/\*\*

\*

\* @param mapNo

\* @return

\* @throws Exception

\*/

public ResultSet getGraphForBean(String mapNo) throws Exception {

String sql = "select mapId MAP\_NO, mapName MAP\_NAME ,mapURL MAP\_URL,"

+ "mapx0,mapy0,mapx1,mapy1,centerx,centery,maplevel,minlevel,maxlevel from map where mapId=?";

return DaoUtil.queryData(sql, new Object[] { mapNo });

}

/\*\*

\* 区域查询

\*/

public ResultSet getAreaInfo(String areaNo) throws Exception {

String sql = "select `ID` , `AREA\_NO` , `AREA\_NAME` ,"

+ "`AREA\_ADDR` , `BUILD\_NUM` , `AREA\_TYPE` ,"

+ "`P\_AREA\_NO` , `IMG\_URL` from d\_area where `AREA\_NO`=?";

return DaoUtil.queryData(sql, new Object[] { areaNo });

}

/\*\*

\* 得到全部不重复的建筑

\*/

public ResultSet getAllBuilding() throws Exception {

String sql = "select `AREA\_NO`, `BUILDING\_NO` BUILD\_NO , `JD` , `WD` from d\_building";

return DaoUtil.queryData(sql, new Object[] {});

}

/\*\*

\* 楼宇查询

\*

\* @param areaNo

\* @param buildNo

\*/

public ResultSet getBuildInfo(String areaNo, String buildNo)

throws Exception {

String sql = "select `ID` , " + "`BUILDING\_NO` BUILD\_NO ,"

+ "`AREA\_NO` ," + "`BUILDING\_NAME` ," + "`BUILDING\_DESC` ,"

+ "`FLOORS` ," + "`FLOOR\_PIX` ," + "`JD` ," + " `WD` ,"

+ "`IMG\_URL` from d\_building where AREA\_NO=? ";

List list = new ArrayList();

list.add(areaNo);

if (!StringUtils.isNullOrBank(buildNo)) {

sql += " and BUILDING\_NO=? ";

list.add(buildNo);

}

return DaoUtil.queryData(sql, list.toArray());

}

/\*\*

\* 楼层查询

\*/

public ResultSet getFloorInfo(String areaNo, String buildNo, String floorNo)

throws Exception {

String sql = " select `ID`, "

+ "`FLOOR\_NO` , "

+ "`BUILD\_NO` , "

+ "`AREA\_NO`, "

+ "`IMG\_URL`, "

+ "`DOORS` , "

+ "`MAP\_NO` from d\_floor where `AREA\_NO`=? and d\_floor.BUILD\_NO=? ";

List list = new ArrayList();

list.add(areaNo);

list.add(buildNo);

if (!StringUtils.isNullOrBank(floorNo)) {

sql += " and d\_floor.FLOOR\_NO=? ";

list.add(floorNo);

}

return DaoUtil.queryData(sql, list.toArray());

}

/\*\*

\* 5.1.5.4.1 房间信息获取请求

\*/

public ResultSet getDoorInfo(String areaNo, String buildNo, String floorNo,

String doorNo) throws Exception {

String sql = "select `ID`, " + "`DOOR\_NO` , " + "`FLOOR\_NO` , "

+ "`BUILD\_NO` , " + "`AREA\_NO` from d\_door "

+ "where `AREA\_NO`=? and `BUILD\_NO`=? and `FLOOR\_NO`=? ";

List list = new ArrayList();

list.add(areaNo);

list.add(buildNo);

list.add(floorNo);

if (!StringUtils.isNullOrBank(doorNo)) {

sql += " and DOOR\_NO=? ";

list.add(doorNo);

}

return DaoUtil.queryData(sql, list.toArray());

}

}

package org.icscn.server.standalone.net.dao.msg;

import java.sql.ResultSet;

import java.util.ArrayList;

import java.util.List;

import org.icscn.server.standalone.db.DaoUtil;

import org.icscn.server.standalone.net.model.msg.GetGroupMsgReqMessage;

import org.icscn.server.standalone.net.model.msg.SendGroupMsgReqMessage;

import org.icscn.server.standalone.net.model.msg.SengSingleMsgReqMessage;

import com.aiyc.framework.utils.UID;

public class MsgDao {

public ResultSet getGroupMsg(String groupId, String rcver) throws Exception {

String sql = "SELECT sender\_no, msg\_sort\_code, content\_msg, send\_time\n"

+ "\tFROM t\_group\_rcv\_msg\n"

+ " WHERE t\_group\_rcv\_msg. group\_id = ?\n"

+ "\t\t\t AND EXISTS (SELECT 1\n"

+ "\t\t\t\t\tFROM t\_group\_rcv\_status\n"

+ "\t\t\t\t WHERE t\_group\_rcv\_status. rcv\_id = t\_group\_rcv\_msg.id\n"

+ "\t\t\t\t\t\t\t AND rcv\_no = ?\n"

+ "\t\t\t\t\t\t\t AND is\_rcv = '01')\n"

+ "\n"

+ " order by send\_time";

return DaoUtil.queryData(sql, new Object[] { groupId, rcver });

}

public void arcGroupMsg(GetGroupMsgReqMessage model) throws Exception {

String sql = "UPDATE t\_group\_rcv\_status\n"

+ "\t SET rcv\_date = now(), is\_rcv = '02'\n"

+ " WHERE rcv\_no = ?\n" + "\t\t\t AND rcv\_id IN (SELECT id\n"

+ "\t\t\t\t\t\t\t\t\t\t\t\tFROM t\_group\_rcv\_msg\n"

+ "\t\t\t\t\t\t\t\t\t\t\t WHERE group\_id = ?)";

DaoUtil.executeQuery(sql, new Object[] { model.MESSAGE.RCVER\_NO,

model.MESSAGE.GROUP\_ID });

// //查看接收消息的数量是否完成

sql = "select count(1) num from t\_group\_rcv\_status,t\_group\_rcv\_msg where "

+ " t\_group\_rcv\_status.rcv\_id= t\_group\_rcv\_msg.id and "

+ "t\_group\_rcv\_status.is\_rcv='01' and t\_group\_rcv\_msg.group\_id=? "

+ "";

ResultSet rs = DaoUtil.queryData(sql,

new Object[] { model.MESSAGE.GROUP\_ID

});

int count = -1;

if (rs.next()) {

count = rs.getInt("num");

}

if (count == 0) {

sql = "INSERT INTO arc\_t\_group\_rcv\_status\n"

+ "\t\t(id, rcv\_id, rcv\_no, is\_rcv,rcv\_date )\n"

+ "\t\tSELECT id, rcv\_id, rcv\_no, is\_rcv,rcv\_date from " +

" t\_group\_rcv\_status WHERE rcv\_id IN (SELECT id\n"

+ "\t\t\t\t\t\t\t\t\t\t\t\t\t\t\t\t\t\t\t\t\t\t\t\t\tFROM t\_group\_rcv\_msg\n"

+ "\t\t\t\t\t\t\t\t\t\t\t\t\t\t\t\t\t\t\t\t\t\t\t\t WHERE group\_id = ?)";

DaoUtil.executeQuery(sql, new Object[] { model.MESSAGE.GROUP\_ID });

sql = "delete from t\_group\_rcv\_status where rcv\_id IN (SELECT id\n"

+ "\t\t\t\t\t\t\t\t\t\t\t\t\t\t\t\t\t\t\t\t\t\t\t\t\tFROM t\_group\_rcv\_msg\n"

+ "\t\t\t\t\t\t\t\t\t\t\t\t\t\t\t\t\t\t\t\t\t\t\t\t WHERE group\_id = ?)";

DaoUtil.executeQuery(sql, new Object[] { model.MESSAGE.GROUP\_ID });

sql = "INSERT INTO arc\_t\_group\_rcv\_msg\n"

+ "\t\t(id, group\_id, sender\_no, msg\_sort\_code, content\_msg, send\_time)\n"

+ "\t\tSELECT id, group\_id, sender\_no, msg\_sort\_code, content\_msg, send\_time\n"

+ "\t\t\tFROM t\_group\_rcv\_msg\n"

+ "\t\t WHERE group\_id = ?";

DaoUtil.executeQuery(sql, new Object[] { model.MESSAGE.GROUP\_ID });

sql = "delete from t\_group\_rcv\_msg where group\_id = ? ";

DaoUtil.executeQuery(sql, new Object[] { model.MESSAGE.GROUP\_ID });

}

}

public void arcMsg(String send, String rcver) throws Exception {

String sql = "INSERT INTO arc\_t\_rcv\_msg\n"

+ "\t\t(id, sender\_no, rcver\_no, msg\_sort\_code, content\_msg, send\_time, rcv\_date)\n"

+ "\t\tSELECT id, sender\_no, rcver\_no, msg\_sort\_code, content\_msg, send\_time, now()\n"

+ "\t\t\tFROM t\_rcv\_msg g\n" + "\t\t WHERE g. sender\_no = ?\n"

+ "\t\t\t\t\t AND g.rcver\_no = ?";

DaoUtil.executeQuery(sql, new Object[] { send, rcver });

sql = " delete from t\_rcv\_msg where sender\_no = ?\n"

+ "\t\t\t\t\t AND rcver\_no = ?";

DaoUtil.executeQuery(sql, new Object[] {send,rcver});

}

public ResultSet getMsg(String send, String rcver) throws Exception {

String sql = "select `MSG\_SORT\_CODE` ,`CONTENT\_MSG`,`SEND\_TIME`"

+ " from t\_rcv\_msg g where g.`SENDER\_NO` =? and g.RCVER\_NO=? order by SEND\_TIME ";

return DaoUtil.queryData(sql, new Object[] { send, rcver });

}

public ResultSet getMsgNum(String rcver) throws Exception {

String sql = "select `SENDER\_NO`,count(1) num from t\_rcv\_msg g where g.RCVER\_NO=? group by SENDER\_NO ";

return DaoUtil.queryData(sql, new Object[] { rcver });

}

public ResultSet getGroupMsgNum(String rcver) throws Exception {

String sql = "select `GROUP\_ID`,count(1) num from `t\_group\_rcv\_msg` g "

+ "where exists (select 1 "

+ " from `t\_group\_rcv\_status`,t\_group\_rcv\_msg where t\_group\_rcv\_status.`RCV\_ID`="

+ "t\_group\_rcv\_msg.id and t\_group\_rcv\_status.`RCV\_NO`=? and "

+ "t\_group\_rcv\_status.`IS\_RCV`='01' ) group by `GROUP\_ID` ";

return DaoUtil.queryData(sql, new Object[] { rcver });

}

public void saveSingleMsg(SengSingleMsgReqMessage model) throws Exception {

String sql = "insert into `t\_rcv\_msg`(\n" + " `SENDER\_NO` ,\n"

+ " `RCVER\_NO`,\n" + " `MSG\_SORT\_CODE` ,\n"

+ " `CONTENT\_MSG` ,\n" + " `SEND\_TIME`)\n"

+ " values(?,?,?,?,now())";

List list = new ArrayList();

list.add(model.MESSAGE.SENDER\_NO);

list.add(model.MESSAGE.RCVER\_NO);

list.add(model.MESSAGE.MSG\_SORT\_CODE);

list.add(model.MESSAGE.CONTENT\_MSG);

DaoUtil.executeQuery(sql, list.toArray());

}

public String saveGroupMsg(SendGroupMsgReqMessage model) throws Exception {

String sql = "insert into `t\_group\_rcv\_msg`(id,\n"

+ " `GROUP\_ID` ,\n" + " `SENDER\_NO` ,\n"

+ " `MSG\_SORT\_CODE` ,\n" + " `CONTENT\_MSG` ,\n"

+ " `SEND\_TIME`\n" + "\n" + " )\n" + " values(?,?,?,?,?,now())";

String id = UID.next();

List list = new ArrayList();

list.add(id);

list.add(model.MESSAGE.GROUP\_ID);

list.add(model.MESSAGE.SENDER\_NO);

list.add(model.MESSAGE.MSG\_SORT\_CODE);

list.add(model.MESSAGE.CONTENT\_MSG);

DaoUtil.executeQuery(sql, list.toArray());

return id;

}

/\*\*

\* 写入群组接收状态

\*/

public void saveGroupMsgStatus(String id, SendGroupMsgReqMessage model)

throws Exception {

String sql = "insert into `t\_group\_rcv\_status`(\n"

+ " `RCV\_ID` ,\n"

+ " `RCV\_NO` ,\n"

+ " `IS\_RCV`\n"

+ "\n"

+ " )\n"

+ "select ?,`USER\_NO`,'01' from `t\_group\_user` where t\_group\_user.`GROUP\_ID`=?" +

" and t\_group\_user.USER\_NO<>? ";

List list = new ArrayList();

list.add(id);

list.add(model.MESSAGE.GROUP\_ID);

list.add(model.MESSAGE.SENDER\_NO);

DaoUtil.executeQuery(sql, list.toArray());

}

}

package org.icscn.server.standalone.net.dao.sell;

import java.sql.ResultSet;

import java.util.ArrayList;

import java.util.HashMap;

import java.util.List;

import java.util.Map;

import org.icscn.server.standalone.db.DaoUtil;

import org.icscn.server.standalone.net.model.seller.SearchCouponReqMessage;

import org.icscn.server.standalone.net.model.seller.SearchDealReqMessage;

import org.icscn.server.standalone.net.model.seller.SearchSjReqMessage;

import com.aiyc.framework.utils.StringUtils;

public class SearchDao {

/\*\*

\* 查询优惠信息

\*/

public ResultSet queryCoupon(SearchCouponReqMessage model) throws Exception {

String sql = "select c.coupon\_id,\n"

+ " c.title,\n"

+ " c. description,\n"

+ " c. regions,\n"

+ " c.categories,\n"

+ " c.download\_count,\n"

+ " c.publish\_date,\n"

+ " c. expiration\_date,\n"

+ " c. logo\_img\_url,\n"

+ " c.coupon\_url,\n"

+ " c. coupon\_h5\_url,\n"

+ " b.area\_no,\n"

+ " b.build\_no,\n"

+ " b.floor\_no,\n"

+ " b.door\_no,\n"

+ " b.business\_id,\n"

+ " sqrt((((? -longitude) \* PI() \* 12656 \*\n"

+ " cos(((? +latitude) / 2) \* PI() / 180) / 180) \*\n"

+ " ((? -longitude) \* PI() \* 12656 \*\n"

+ " cos(((? +latitude) / 2) \* PI() / 180) / 180)) +\n"

+ " (((? -latitude) \* PI() \* 12656 / 180) \*\n"

+ " ((? -latitude) \* PI() \* 12656 / 180))) \* 1000 distance,\n"

+ " b.business\_name\n"

+ " from s\_businesses\_detail a, s\_businesses b, s\_coupon\_detail c, s\_coupon d\n"

+ " where a.business\_id = b.business\_id\n"

+ " and a.business\_id = d.business\_id\n"

+ " and c.coupon\_id = d.coupon\_id";

List list = new ArrayList();

list.add(model.MESSAGE.longitude);

list.add(model.MESSAGE.latitude);

list.add(model.MESSAGE.longitude);

list.add(model.MESSAGE.latitude);

list.add(model.MESSAGE.latitude);

list.add(model.MESSAGE.latitude);

if (!StringUtils.isNullOrBank(model.MESSAGE.coupon\_id)) {

sql += " and c.coupon\_id=? ";

list.add(model.MESSAGE.coupon\_id);

}

if (!StringUtils.isNullOrBank(model.MESSAGE.latitude)) {

sql += " and sqrt(\n"

+ " (\n"

+ " ((?-longitude)\*PI()\*12656\*cos(((?+latitude)/2)\*PI()/180)/180)\n"

+ " \*\n"

+ " ((?-longitude)\*PI()\*12656\*cos (((?+latitude)/2)\*PI()/180)/180)\n"

+ " )\n" + " +\n" + " (\n"

+ " ((?-latitude)\*PI()\*12656/180)\n" + " \*\n"

+ " ((?-latitude)\*PI()\*12656/180)\n" + " )\n"

+ ")<?/1000";

list.add(model.MESSAGE.longitude);

list.add(model.MESSAGE.latitude);

list.add(model.MESSAGE.longitude);

list.add(model.MESSAGE.latitude);

list.add(model.MESSAGE.latitude);

list.add(model.MESSAGE.latitude);

list.add(model.MESSAGE.radius);

}

if (!StringUtils.isNullOrBank(model.MESSAGE.area\_no)) {

sql += " and b.area\_no=? ";

list.add(model.MESSAGE.area\_no);

}

if (!StringUtils.isNullOrBank(model.MESSAGE.build\_no)) {

sql += " and b.build\_no=? ";

list.add(model.MESSAGE.build\_no);

}

if (!StringUtils.isNullOrBank(model.MESSAGE.floor\_no)) {

sql += " and b.floor\_no=? ";

list.add(model.MESSAGE.floor\_no);

}

if (!StringUtils.isNullOrBank(model.MESSAGE.door\_no)) {

sql += " and b.door\_no=? ";

list.add(model.MESSAGE.door\_no);

}

if (!StringUtils.isNullOrBank(model.MESSAGE.region)) {

sql += " and c.regions like CONCAT('%',?,'%') ";

list.add(model.MESSAGE.region);

}

if (!StringUtils.isNullOrBank(model.MESSAGE.category)) {

sql += " and c.categories like CONCAT('%',?,'%') ";

list.add(model.MESSAGE.category);

}

if (!StringUtils.isNullOrBank(model.MESSAGE.keyword)) {

sql += " and (c.title like CONCAT('%',?,'%') or c.description like CONCAT('%',?,'%') ) ";

list.add(model.MESSAGE.keyword);

list.add(model.MESSAGE.keyword);

}

int n1 = (Integer.parseInt(model.MESSAGE.page) - 1)

\* Integer.parseInt(model.MESSAGE.limit);

int n2 = Integer.parseInt(model.MESSAGE.page)

\* Integer.parseInt(model.MESSAGE.limit);

String sort = model.MESSAGE.sort;

// 1:默认，

// 2:星级高优先，

// 3:产品评价高优先，

// 4:环境评价高优先，

// 5:服务评价高优先，

// 6:点评数量多优先，

// 7:离传入经纬度坐标距离近优先，

// 8:人均价格低优先，

// 9：人均价格高优先

Map map = new HashMap();

map.put("2", "avg\_rating desc");

map.put("3", "product\_grade desc ");

map.put("4", "decoration\_grade desc");

map.put("5", "service\_grade desc ");

map.put("6", "product\_score desc");

map.put("7", "distance desc");

map.put("8", "avg\_price asc");

map.put("9", "avg\_price desc ");

String s = (String) map.get(sort);

if (s != null && !"".equals(s)) {

sql += " order by " + s + " ";

}

sql += " limit ?,? ";

list.add(n1);

list.add(n2);

return DaoUtil.queryData(sql, list.toArray());

}

/\*\*

\* 查询团购信息

\*/

public ResultSet queryDeals(String bsId) throws Exception {

String sql = " select `coupon\_id` id , `coupon\_description` description"

+ " , `coupon\_url` url from `s\_coupon`"

+ " where business\_id=? ";

ResultSet rs = DaoUtil.queryData(sql, new Object[] { bsId });

return rs;

}

/\*\*

\* 查询

\*/

public ResultSet querySJ(SearchSjReqMessage model) throws Exception {

String sql =

"select a.business\_id,\n"

+ " name,\n"

+ " branch\_name,\n"

+ " address,\n"

+ " city,\n"

+ " telephone,\n"

+ " latitude,\n"

+ " longitude,\n"

+ " avg\_rating,\n"

+ " rating\_img\_url,\n"

+ " rating\_s\_img\_url,\n"

+ " product\_grade,\n"

+ " decoration\_grade,\n"

+ " service\_grade,\n"

+ " product\_score,\n"

+ " decoration\_score,\n"

+ " service\_score,\n"

+ " avg\_price,\n"

+ " review\_count,\n"

+ " review\_list\_url,\n"

+ " a.business\_url,\n"

+ " photo\_url,\n"

+ " s\_photo\_url,\n"

+ " photo\_count,\n"

+ " photo\_list\_url,\n"

+ " has\_coupon,\n"

+ " coupon\_id,\n"

+ " coupon\_description,\n"

+ " coupon\_url,\n"

+ " has\_deal,\n"

+ " deal\_count,\n"

+ " has\_online\_reservation,\n"

+ " online\_reservation\_url,\n"

+ " regions,\n"

+ " categories,\n"

+ " b.area\_no,\n"

+ " b.build\_no,\n"

+ " b.floor\_no,\n"

+ " b.door\_no, sqrt(\n"

+ " (\n"

+ " ((?-longitude)\*PI()\*12656\*cos(((?+latitude)/2)\*PI()/180)/180)\n"

+ " \*\n"

+ " ((?-longitude)\*PI()\*12656\*cos (((?+latitude)/2)\*PI()/180)/180)\n"

+ " )\n" + " +\n" + " (\n"

+ " ((?-latitude)\*PI()\*12656/180)\n" + " \*\n"

+ " ((?-latitude)\*PI()\*12656/180)\n" + " )\n"

+ ")\*1000 distance \n"

+ " from s\_businesses\_detail a, s\_businesses b\n"

+ " where a.business\_id = b.business\_id";

List list = new ArrayList();

list.add(model.MESSAGE.longitude);

list.add(model.MESSAGE.latitude);

list.add(model.MESSAGE.longitude);

list.add(model.MESSAGE.latitude);

list.add(model.MESSAGE.latitude);

list.add(model.MESSAGE.latitude);

if (!StringUtils.isNullOrBank(model.MESSAGE.business\_id)) {

sql += " and b.business\_id=? ";

list.add(model.MESSAGE.business\_id);

}

if (!StringUtils.isNullOrBank(model.MESSAGE.latitude)) {

sql += " and sqrt(\n"

+ " (\n"

+ " ((?-longitude)\*PI()\*12656\*cos(((?+latitude)/2)\*PI()/180)/180)\n"

+ " \*\n"

+ " ((?-longitude)\*PI()\*12656\*cos (((?+latitude)/2)\*PI()/180)/180)\n"

+ " )\n" + " +\n" + " (\n"

+ " ((?-latitude)\*PI()\*12656/180)\n" + " \*\n"

+ " ((?-latitude)\*PI()\*12656/180)\n" + " )\n"

+ ")<?/1000";

list.add(model.MESSAGE.longitude);

list.add(model.MESSAGE.latitude);

list.add(model.MESSAGE.longitude);

list.add(model.MESSAGE.latitude);

list.add(model.MESSAGE.latitude);

list.add(model.MESSAGE.latitude);

list.add(model.MESSAGE.radius);

}

if (!StringUtils.isNullOrBank(model.MESSAGE.area\_no)) {

sql += " and b.area\_no=? ";

list.add(model.MESSAGE.area\_no);

}

if (!StringUtils.isNullOrBank(model.MESSAGE.build\_no)) {

sql += " and b.build\_no=? ";

list.add(model.MESSAGE.build\_no);

}

if (!StringUtils.isNullOrBank(model.MESSAGE.floor\_no)) {

sql += " and b.floor\_no=? ";

list.add(model.MESSAGE.floor\_no);

}

if (!StringUtils.isNullOrBank(model.MESSAGE.door\_no)) {

sql += " and b.door\_no=? ";

list.add(model.MESSAGE.door\_no);

}

if (!StringUtils.isNullOrBank(model.MESSAGE.region)) {

sql += " and a.regions like CONCAT('%',?,'%') ";

list.add(model.MESSAGE.region);

}

if (!StringUtils.isNullOrBank(model.MESSAGE.category)) {

sql += " and a.categories like CONCAT('%',?,'%') ";

list.add(model.MESSAGE.category);

}

if (!StringUtils.isNullOrBank(model.MESSAGE.keyword)) {

sql += " and (a.name like CONCAT('%',?,'%') or a.address like CONCAT('%',?,'%') ) ";

list.add(model.MESSAGE.keyword);

list.add(model.MESSAGE.keyword);

}

if (!StringUtils.isNullOrBank(model.MESSAGE.has\_coupon)) {

sql += " and a.has\_coupon =? ";

list.add(model.MESSAGE.has\_coupon);

}

if (!StringUtils.isNullOrBank(model.MESSAGE.has\_deal)) {

sql += " and a.has\_deal =? ";

list.add(model.MESSAGE.has\_deal);

}

if (!StringUtils.isNullOrBank(model.MESSAGE.has\_online\_reservation)) {

sql += " and a.has\_online\_reservation =? ";

list.add(model.MESSAGE.has\_online\_reservation);

}

int n1 = (Integer.parseInt(model.MESSAGE.page) - 1)

\* Integer.parseInt(model.MESSAGE.limit);

int n2 = Integer.parseInt(model.MESSAGE.page)

\* Integer.parseInt(model.MESSAGE.limit);

String sort = model.MESSAGE.sort;

// 1:默认，

// 2:星级高优先，

// 3:产品评价高优先，

// 4:环境评价高优先，

// 5:服务评价高优先，

// 6:点评数量多优先，

// 7:离传入经纬度坐标距离近优先，

// 8:人均价格低优先，

// 9：人均价格高优先

Map map = new HashMap();

map.put("2", "avg\_rating desc");

map.put("3", "product\_grade desc ");

map.put("4", "decoration\_grade desc");

map.put("5", "service\_grade desc ");

map.put("6", "product\_score desc");

map.put("7", "distance desc");

map.put("8", "avg\_price asc");

map.put("9", "avg\_price desc ");

String s = (String) map.get(sort);

if (s != null && !"".equals(s)) {

sql += " order by " + s + " ";

}

sql += " limit ?,? ";

list.add(n1);

list.add(n2);

return DaoUtil.queryData(sql, list.toArray());

}

/\*\*

\* 查询团购信息

\*/

public ResultSet queryDeal(SearchDealReqMessage model) throws Exception {

String sql = "select c.deal\_id,\n"

+ " title,\n"

+ " description,\n"

+ " list\_price,\n"

+ " current\_price,\n"

+ " c.regions ,\n"

+ " c.categories,\n"

+ " purchase\_count,\n"

+ " publish\_date,\n"

+ " purchase\_deadline,\n"

+ " image\_url,\n"

+ " s\_image\_url,\n"

+ " deal\_url,\n"

+ " deal\_h5\_url,\n"

+ " commission\_ratio,\n"

+ " sqrt((((? -longitude) \* PI() \* 12656 \*\n"

+ " cos(((? +latitude) / 2) \* PI() / 180) / 180) \*\n"

+ " ((? -longitude) \* PI() \* 12656 \*\n"

+ " cos(((? +latitude) / 2) \* PI() / 180) / 180)) +\n"

+ " (((? -latitude) \* PI() \* 12656 / 180) \*\n"

+ " ((? -latitude) \* PI() \* 12656 / 180))) \* 1000 distance,\n"

+ " b.area\_no,\n"

+ " b.build\_no,\n"

+ " b.floor\_no,\n"

+ " b.door\_no,\n"

+ " b.business\_name,\n"

+ " b.business\_id,longitude,latitude\n"

+ "\n"

+ " from s\_businesses\_detail a, s\_businesses b, s\_deals\_detail c, s\_deals d\n"

+ " where a.business\_id = b.business\_id\n"

+ " and a.business\_id = d.business\_id\n"

+ " and c.deal\_id = d.deal\_id";

List list = new ArrayList();

list.add(model.MESSAGE.longitude);

list.add(model.MESSAGE.latitude);

list.add(model.MESSAGE.longitude);

list.add(model.MESSAGE.latitude);

list.add(model.MESSAGE.latitude);

list.add(model.MESSAGE.latitude);

if (!StringUtils.isNullOrBank(model.MESSAGE.deal\_id)) {

sql += " and c.deal\_id=? ";

list.add(model.MESSAGE.deal\_id);

}

if (!StringUtils.isNullOrBank(model.MESSAGE.latitude)) {

sql += " and sqrt(\n"

+ " (\n"

+ " ((?-longitude)\*PI()\*12656\*cos(((?+latitude)/2)\*PI()/180)/180)\n"

+ " \*\n"

+ " ((?-longitude)\*PI()\*12656\*cos (((?+latitude)/2)\*PI()/180)/180)\n"

+ " )\n" + " +\n" + " (\n"

+ " ((?-latitude)\*PI()\*12656/180)\n" + " \*\n"

+ " ((?-latitude)\*PI()\*12656/180)\n" + " )\n"

+ ")<?/1000";

list.add(model.MESSAGE.longitude);

list.add(model.MESSAGE.latitude);

list.add(model.MESSAGE.longitude);

list.add(model.MESSAGE.latitude);

list.add(model.MESSAGE.latitude);

list.add(model.MESSAGE.latitude);

list.add(model.MESSAGE.radius);

}

if (!StringUtils.isNullOrBank(model.MESSAGE.area\_no)) {

sql += " and b.area\_no=? ";

list.add(model.MESSAGE.area\_no);

}

if (!StringUtils.isNullOrBank(model.MESSAGE.build\_no)) {

sql += " and b.build\_no=? ";

list.add(model.MESSAGE.build\_no);

}

if (!StringUtils.isNullOrBank(model.MESSAGE.floor\_no)) {

sql += " and b.floor\_no=? ";

list.add(model.MESSAGE.floor\_no);

}

if (!StringUtils.isNullOrBank(model.MESSAGE.door\_no)) {

sql += " and b.door\_no=? ";

list.add(model.MESSAGE.door\_no);

}

if (!StringUtils.isNullOrBank(model.MESSAGE.region)) {

sql += " and c.regions like CONCAT('%',?,'%') ";

list.add(model.MESSAGE.region);

}

if (!StringUtils.isNullOrBank(model.MESSAGE.category)) {

sql += " and c.categories like CONCAT('%',?,'%') ";

list.add(model.MESSAGE.category);

}

if (!StringUtils.isNullOrBank(model.MESSAGE.keyword)) {

sql += " and (c.title like CONCAT('%',?,'%') or c.description like CONCAT('%',?,'%') ) ";

list.add(model.MESSAGE.keyword);

list.add(model.MESSAGE.keyword);

}

int n1 = (Integer.parseInt(model.MESSAGE.page) - 1)

\* Integer.parseInt(model.MESSAGE.limit);

int n2 = Integer.parseInt(model.MESSAGE.page)

\* Integer.parseInt(model.MESSAGE.limit);

String sort = model.MESSAGE.sort;

// 1:默认，

// 2:星级高优先，

// 3:产品评价高优先，

// 4:环境评价高优先，

// 5:服务评价高优先，

// 6:点评数量多优先，

// 7:离传入经纬度坐标距离近优先，

// 8:人均价格低优先，

// 9：人均价格高优先

Map map = new HashMap();

map.put("2", "avg\_rating desc");

map.put("3", "product\_grade desc ");

map.put("4", "decoration\_grade desc");

map.put("5", "service\_grade desc ");

map.put("6", "product\_score desc");

map.put("7", "distance desc");

map.put("8", "avg\_price asc");

map.put("9", "avg\_price desc ");

String s = (String) map.get(sort);

if (s != null && !"".equals(s)) {

sql += " order by " + s + " ";

}

sql += " limit ?,? ";

list.add(n1);

list.add(n2);

return DaoUtil.queryData(sql, list.toArray());

}

}

package org.icscn.server.standalone.net.dao.sell;

import java.sql.ResultSet;

import java.util.ArrayList;

import java.util.Iterator;

import java.util.List;

import org.icscn.server.standalone.db.DaoUtil;

import org.icscn.server.standalone.net.dao.area.GraphDao;

import org.icscn.server.standalone.net.dao.comm.CommDao;

import org.icscn.server.standalone.net.model.seller.AddFreeReqMessage;

import org.icscn.server.standalone.net.model.seller.GetProductRepMessage;

import org.icscn.server.standalone.net.model.seller.GetSellerFreeRepMessage;

import org.icscn.server.standalone.net.model.seller.PublishProductReqMessage;

import org.icscn.server.standalone.net.model.seller.SellerJoinReqMessage;

import com.aiyc.framework.utils.DateUtils;

import com.aiyc.framework.utils.StringUtils;

import com.aiyc.framework.utils.UID;

public class SellerDao {

private GraphDao dao = new GraphDao();

private CommDao cdao = new CommDao();

public void saveFree(AddFreeReqMessage model) throws Exception {

String sjNo = model.MESSAGE.SJ\_NO;

String sql = "insert into s\_hd\n" + "( `SJ\_ID` ,\n"

+ " `HD\_NAME` ,\n" + " `DETAIL`,\n" + " `SDATE` ,\n"

+ " `EDATE` ,\n" + " `HB` )\n" + "\n" + " select id , "

+ DaoUtil.makePix(5) + " from s\_seller s where s.sj\_no=?";

Iterator<GetSellerFreeRepMessage> it = model.MESSAGE.FREE\_LIST

.iterator();

int index = 1;

while (it.hasNext()) {

GetSellerFreeRepMessage next = it.next();

List list = new ArrayList();

list.add(next.HD\_NAME);

list.add(next.DETAIL);

list.add(next.SDATE);

list.add(next.EDATE);

list.add(next.HB);

DaoUtil.executeQuery(sql, list.toArray());

}

}

/\*\*

\* 获取优惠

\*/

public ResultSet getSellerFree(String sjNo, String key) throws Exception {

String sql = "select\n" + " s\_seller.SJ\_NO ,\n" + " `HD\_NAME`,\n"

+ " `DETAIL`,\n" + " `SDATE`,\n" + " `EDATE` ,\n"

+ " `HB`,'' RTN\_FLAG,'' RTN\_MSG from s\_hd ,s\_seller where\n"

+ "\n" + " s\_hd.sj\_id=s\_seller.id\n" + "";

List list = new ArrayList();

if (!StringUtils.isNullOrBank(sjNo)) {

sql += " and s\_seller.SJ\_NO=? ";

list.add(sjNo);

}

if (!StringUtils.isNullOrBank(key)) {

sql += " and DETAIL like CONCAT('%',?,'%') ";

list.add(key);

}

return DaoUtil.queryData(sql, list.toArray());

}

/\*\*

\* 添加商品

\*/

public void saveProduct(PublishProductReqMessage model) throws Exception {

String sjNo = model.MESSAGE.SJ\_NO;

Iterator<GetProductRepMessage> it = model.MESSAGE.CP\_LIST.iterator();

String sql = "insert into `s\_product` (\n" + "`SJ\_ID` ,\n"

+ "`CP\_NO` ,\n" + "`CP\_NAME` ,\n" + "`CP\_MODEL` ,\n"

+ "`CP\_DES` ,\n" + "`CP\_SORT` ,\n" + "`UP\_DATE` ,\n"

+ "`DOWN\_DATE` ,\n" + "`PRICE` ,\n" + "`RATE` ,\n"

+ "`TOTAL` ,\n" + "`PRE\_NUM` ,\n" + "`STATUS`,\n"

+ "`CP\_KEY`)\n" + "select sss.id, " + DaoUtil.makePix(13)

+ " from s\_seller sss where sj\_no=?";

while (it.hasNext()) {

GetProductRepMessage next = it.next();

DaoUtil.executeQuery("delete from s\_product where cp\_no=? ",

new Object[] { next.CP\_NO });

List list = new ArrayList();

String cpNo = next.CP\_NO==null||"".equals(next.CP\_NO)?UID.next():next.CP\_NO

;

list.add(cpNo);

list.add(next.CP\_NAME);

list.add(next.CP\_MODEL);

list.add(next.CP\_DES);

list.add(next.CP\_SORT);

if (next.UP\_DATE == null || "".equals(next.UP\_DATE)) {

list.add(DateUtils.getCurrentDateStr());

} else {

list.add(next.UP\_DATE);

}

if (next.DOWN\_DATE == null || "".equals(next.DOWN\_DATE)) {

list.add(DateUtils.getCurrentDateStr());

} else {

list.add(next.DOWN\_DATE);

}

list.add(next.PRICE);

list.add(next.RATE);

list.add(next.TOTAL);

list.add(next.PRE\_NUM);

list.add("01");

list.add(next.CP\_KEY);

list.add(sjNo);

DaoUtil.executeQuery(sql, list.toArray());

}

}

/\*\*

\* 商家入驻

\*/

public String saveSeller(SellerJoinReqMessage model) throws Exception {

String sql = "INSERT INTO s\_seller\n"

+ "\t\t( sj\_no, sj\_name, sj\_key, jy\_sort, yy\_sdate, yy\_edate, description, area\_no,\n"

+ "\t\t build\_no, floor\_no, door\_no, status, zz\_no)\n"

+ " values(" + DaoUtil.makePix(13) + ")";

List list = new ArrayList();

String sjNo = UID.next();

list.add(sjNo);

list.add(model.MESSAGE.SJ\_NAME);

list.add(model.MESSAGE.KEY);

list.add(model.MESSAGE.JY\_SORT);

if (model.MESSAGE.YY\_SDATE == null || "".equals(model.MESSAGE.YY\_SDATE)) {

list.add(DateUtils.getCurrentDateStr());

} else {

list.add(model.MESSAGE.YY\_SDATE);

}

if (model.MESSAGE.YY\_EDATE == null || "".equals(model.MESSAGE.YY\_EDATE)) {

list.add(DateUtils.getCurrentDateStr());

} else {

list.add(model.MESSAGE.YY\_EDATE);

}

list.add(model.MESSAGE.DESCRIPTION);

list.add(model.MESSAGE.AREA\_NO);

list.add(model.MESSAGE.BUILD\_NO);

list.add(model.MESSAGE.FLOOR\_NO);

list.add(model.MESSAGE.DOOR\_NO);

list.add("01");

list.add(model.MESSAGE.USER\_NO);

DaoUtil.executeQuery(sql, list.toArray());

return sjNo;

}

/\*\*

\* 商家信息修改

\*/

public String editSeller(SellerJoinReqMessage model) throws Exception {

String sql = "update s\_seller set sj\_name=?, sj\_key=?, jy\_sort=?, yy\_sdate=?"

+ ", yy\_edate=?, description=? ";

List list = new ArrayList();

list.add(model.MESSAGE.SJ\_NAME);

list.add(model.MESSAGE.KEY);

if (model.MESSAGE.YY\_SDATE == null || "".equals(model.MESSAGE.YY\_SDATE)) {

list.add(DateUtils.getCurrentDateStr());

} else {

list.add(model.MESSAGE.YY\_SDATE);

}

if (model.MESSAGE.YY\_EDATE == null || "".equals(model.MESSAGE.YY\_EDATE)) {

list.add(DateUtils.getCurrentDateStr());

} else {

list.add(model.MESSAGE.YY\_EDATE);

}

list.add(model.MESSAGE.DESCRIPTION);

if (!StringUtils.isNullOrBank(model.MESSAGE.AREA\_NO)) {

sql += " ,area\_no=? ";

list.add(model.MESSAGE.AREA\_NO);

}

if (!StringUtils.isNullOrBank(model.MESSAGE.BUILD\_NO)) {

sql += " ,build\_no=? ";

list.add(model.MESSAGE.BUILD\_NO);

}

if (!StringUtils.isNullOrBank(model.MESSAGE.FLOOR\_NO)) {

sql += " ,floor\_no=? ";

list.add(model.MESSAGE.FLOOR\_NO);

}

if (!StringUtils.isNullOrBank(model.MESSAGE.DOOR\_NO)) {

sql += " ,door\_no=? ";

list.add(model.MESSAGE.DOOR\_NO);

}

if (!StringUtils.isNullOrBank(model.MESSAGE.JY\_SORT)) {

sql += " ,jy\_sort=? ";

list.add(model.MESSAGE.JY\_SORT);

}

sql += " where sj\_no=? ";

list.add(model.MESSAGE.USER\_NO);

DaoUtil.executeQuery(sql, list.toArray());

return model.MESSAGE.SJ\_NO;

}

/\*\*

\* 查询产品

\*/

public ResultSet getProduct(String sjNo, String key, String cpNo)

throws Exception {

String sql = "SELECT\n" + "s\_seller.SJ\_NO,\n" + "s\_product.CP\_NO,\n"

+ "s\_product.CP\_NAME,\n" + "s\_product.CP\_MODEL,\n"

+ "s\_product.CP\_DES,\n" + "s\_product.CP\_SORT,\n"

+ "s\_product.UP\_DATE,\n" + "s\_product.DOWN\_DATE,\n"

+ "s\_product.PRICE,\n" + "s\_product.RATE,\n"

+ "s\_product.TOTAL,\n" + "s\_product.PRE\_NUM,\n"

+ "s\_product.STATUS,\n"

+ "s\_product.CP\_KEY ,'' RTN\_FLAG,'' RTN\_MSG\n" + "FROM\n"

+ "s\_product,s\_seller\n"

+ " where s\_product.SJ\_ID=s\_seller.id";

List list = new ArrayList();

if (!StringUtils.isNullOrBank(sjNo)) {

sql += " and s\_seller.SJ\_NO=? ";

list.add(sjNo);

}

if (!StringUtils.isNullOrBank(key)) {

sql += " and s\_product.CP\_KEY like CONCAT('%',?,'%') ";

list.add(key);

}

if (!StringUtils.isNullOrBank(cpNo)) {

sql += " and s\_product.CP\_NO =?";

list.add(cpNo);

}

return DaoUtil.queryData(sql, list.toArray());

}

/\*\*

\* 通过关键字和商家编号查询商家

\*/

public ResultSet getSeller(String sjNo, String key) throws Exception {

String sql = "SELECT\n" + "s\_seller.SJ\_NO,\n" + "s\_seller.SJ\_NAME,\n"

+ "s\_seller.SJ\_KEY,\n" + "s\_seller.JY\_SORT,\n"

+ "s\_seller.YY\_SDATE,\n" + "s\_seller.YY\_EDATE,\n"

+ "s\_seller.DESCRIPTION,\n" + "s\_seller.AREA\_NO,\n"

+ "s\_seller.BUILD\_NO,\n" + "s\_seller.FLOOR\_NO,\n"

+ "s\_seller.DOOR\_NO,\n"

+ "s\_seller.ZZ\_NO,STATUS,'' RTN\_FLAG,'' RTN\_MSG\n" + "FROM\n"

+ "s\_seller where STATUS='01' ";

List list = new ArrayList();

if (!StringUtils.isNullOrBank(sjNo)) {

sql += " and SJ\_NO=? ";

list.add(sjNo);

}

if (!StringUtils.isNullOrBank(key)) {

sql += " and SJ\_KEY like CONCAT('%',?,'%') ";

list.add(key);

}

if (list.size() == 0) {

return null;

}

return DaoUtil.queryData(sql, list.toArray());

}

public ResultSet getSeller(String areaNo, String buildNo, String floorNo,

String doorNo) throws Exception {

String sql = "select " + "`ID`, " + "`SJ\_NO` , " + " `SJ\_NAME` , "

+ " `SJ\_KEY`, " + " `JY\_SORT` , " + " `YY\_SDATE` , "

+ " `YY\_EDATE` , " + " `DESCRIPTION` , " + " `AREA\_NO` , "

+ " `BUILD\_NO` , " + " `FLOOR\_NO`, " + " `DOOR\_NO` "

+ "from s\_seller where s\_seller.AREA\_NO=? "

+ "and s\_seller.BUILD\_NO=? and s\_seller.FLOOR\_NO=? "

+ "and s\_seller.DOOR\_NO=? ";

return DaoUtil.queryData(sql, new Object[] { areaNo, buildNo, floorNo,

doorNo });

}

}

package org.icscn.server.standalone.util;

import java.lang.annotation.Annotation;

import java.util.ArrayList;

import java.util.Hashtable;

import java.util.List;

import java.util.Map;

import org.icscn.server.standalone.net.IHandler;

import com.aiyc.framework.annotation.Requestable;

public final class ConsoleData {

public static final String HNADER = "hander";

private static class ConsoleDataImpl {

static ConsoleData instance = new ConsoleData();

}

private ConsoleData() {

data = new Hashtable();

init();

}

public static ConsoleData instance() {

return ConsoleDataImpl.instance;

}

public Map data = null;

public void addCached(Object objectName, Object value) {

data.put(objectName, value);

}

public Object getCachedValue(Object name) {

return data.get(name);

}

private void init() {

if (data.get(HNADER) != null)

return;

List<String> classes = new ArrayList<String>();

String patsh[] = Configuration.handers.split(";");

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

try {

classes.addAll(ClassUtil.getClassName(patsh[i]));

} catch (Exception e) {

e.printStackTrace();

}

}

Map handers = new Hashtable(1024);

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

try {

String path = classes.get(i);

Class handClass = Class.forName(path);

Object hander = handClass.newInstance();

if (hander instanceof IHandler) {

IHandler hand = (IHandler) hander;

Annotation ano = handClass.getAnnotation(Requestable.class);

if (ano != null) {

Requestable req = (Requestable) ano;

handers.put(req.serverName(), hand);

}

}

} catch (java.lang.InstantiationException e) {

}

catch (Exception e) {

e.printStackTrace();

}

}

data.put(HNADER, handers);

}

}

package org.icscn.server.standalone.util;

import java.io.File;

import java.net.URL;

import java.net.URLClassLoader;

import java.util.ArrayList;

import java.util.Enumeration;

import java.util.List;

import java.util.jar.JarEntry;

import java.util.jar.JarFile;

public class ClassUtil {

/\*\*

\* 获取某包下（包括该包的所有子包）所有类

\*

\* @param packageName

\* 包名

\* @return 类的完整名称

\*/

public static List<String> getClassName(String packageName)

throws Exception {

return getClassName(packageName, true);

}

/\*\*

\* 获取某包下所有类

\*

\* @param packageName

\* 包名

\* @param childPackage

\* 是否遍历子包

\* @return 类的完整名称

\*/

public static List<String> getClassName(String packageName,

boolean childPackage) throws Exception {

List<String> fileNames = null;

ClassLoader loader = Thread.currentThread().getContextClassLoader();

String packagePath = packageName.replace(".", "/");

URL url = loader.getResource(packagePath);

if (url != null) {

String type = url.getProtocol();

if (type.equals("file")) {

fileNames = getClassNameByFile(packageName, java.net.URLDecoder

.decode(url.getPath(), "UTF-8"), null, childPackage);

} else if (type.equals("jar")) {

fileNames = getClassNameByJar(java.net.URLDecoder

.decode(packageName,"UTF-8"), url.getPath(),

childPackage);

}

} else {

fileNames = getClassNameByJars(packageName,

((URLClassLoader) loader).getURLs(), packagePath,

childPackage);

}

return fileNames;

}

/\*\*

\* 从项目文件获取某包下所有类

\*

\* @param filePath

\* 文件路径

\* @param className

\* 类名集合

\* @param childPackage

\* 是否遍历子包

\* @return 类的完整名称

\*/

private static List<String> getClassNameByFile(String packName,

String filePath, List<String> className, boolean childPackage) {

List<String> myClassName = new ArrayList<String>();

File file = new File(filePath);

File[] childFiles = file.listFiles();

for (File childFile : childFiles) {

if (childFile.isDirectory()) {

if (childPackage) {

myClassName.addAll(getClassNameByFile(packName, childFile

.getPath(), myClassName, childPackage));

}

} else {

String childFilePath = childFile.getPath();

if (childFilePath.endsWith(".class")) {

childFilePath = childFilePath.substring(childFilePath

.indexOf("\\classes") + 9, childFilePath

.lastIndexOf("."));

childFilePath = childFilePath.replace("\\", ".");

childFilePath = childFilePath.substring(childFilePath

.indexOf(packName) );

myClassName.add(childFilePath);

}

}

}

return myClassName;

}

/\*\*

\* 从jar获取某包下所有类

\*

\* @param jarPath

\* jar文件路径

\* @param childPackage

\* 是否遍历子包

\* @return 类的完整名称

\*/

private static List<String> getClassNameByJar(String packName,

String jarPath, boolean childPackage) {

List<String> myClassName = new ArrayList<String>();

String[] jarInfo = jarPath.split("!");

String jarFilePath = jarInfo[0].substring(jarInfo[0].indexOf("/"));

String packagePath = jarInfo[1].substring(1);

try {

JarFile jarFile = new JarFile(jarFilePath);

Enumeration<JarEntry> entrys = jarFile.entries();

while (entrys.hasMoreElements()) {

JarEntry jarEntry = entrys.nextElement();

String entryName = jarEntry.getName();

if (entryName.endsWith(".class")) {

if (childPackage) {

if (entryName.startsWith(packagePath)) {

entryName = entryName.replace("/", ".").substring(

0, entryName.lastIndexOf("."));

entryName = entryName.substring(entryName

.indexOf(packName) );

myClassName.add(entryName);

}

} else {

int index = entryName.lastIndexOf("/");

String myPackagePath;

if (index != -1) {

myPackagePath = entryName.substring(0, index);

} else {

myPackagePath = entryName;

}

if (myPackagePath.equals(packagePath)) {

entryName = entryName.replace("/", ".").substring(

0, entryName.lastIndexOf("."));

entryName = entryName.substring(entryName

.indexOf(packName) );

myClassName.add(entryName);

}

}

}

}

} catch (Exception e) {

e.printStackTrace();

}

return myClassName;

}

/\*\*

\* 从所有jar中搜索该包，并获取该包下所有类

\*

\* @param urls

\* URL集合

\* @param packagePath

\* 包路径

\* @param childPackage

\* 是否遍历子包

\* @return 类的完整名称

\*/

private static List<String> getClassNameByJars(String packNam, URL[] urls,

String packagePath, boolean childPackage) {

List<String> myClassName = new ArrayList<String>();

if (urls != null) {

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

URL url = urls[i];

String urlPath = url.getPath();

// 不必搜索classes文件夹

if (urlPath.endsWith("classes/")) {

continue;

}

String jarPath = urlPath + "!/" + packagePath;

myClassName.addAll(getClassNameByJar(packNam, jarPath,

childPackage));

}

}

return myClassName;

}

}

package org.icscn.server.standalone.net.model.seller;

import org.icscn.server.standalone.net.model.BaseObject;

public class SearchSjReqMessage extends BaseObject {

public Message MESSAGE = new Message();

public static class Message {

public String business\_id;// 商户ID int 商户ID

public String latitude;// 纬度坐标 float 纬度坐标，须与经度坐标同时传入，与城市名称二者必选其一传入

public String longitude;// 经度坐标 float 经度坐标，须与纬度坐标同时传入，与城市名称二者必选其一传入

public String radius;// 搜索半径 int 搜索半径，单位为米，最小值1，最大值5000，如不传入默认为1000

public String area\_no;// 区域编号 string 区域编号(经纬度没有传入的情况下必须传入)

public String build\_no;// 建筑编号 string 建筑编号

public String floor\_no;// 楼层编号 string 楼层编号

public String door\_no;// 房间编号 string 房间编号

public String region;// 城市区域名 string

// 城市区域名，可选范围见相关API返回结果（不含返回结果中包括的城市名称信息），如传入城市区域名，则城市名称必须传入

public String category;// 分类名 string

// 分类名，可选范围见相关API返回结果；支持同时输入多个分类，以逗号分隔，最大不超过5个。

public String keyword;// 关键词 string 关键词，搜索范围包括商户名、地址、标签等

public String out\_offset\_type;// 传出经纬度偏移类型1 int

// 传出经纬度偏移类型，1:高德坐标系偏移，2:图吧坐标系偏移，如不传入，默认值为1

public String platform;// 传出链接类型 int

// 传出链接类型，1:web站链接（适用于网页应用），2:HTML5站链接（适用于移动应用和联网车载应用），如不传入，默认值为1

public String has\_coupon;// 是否有优惠券 int 根据是否有优惠券来筛选返回的商户，1:有，0:没有

public String has\_deal;// 是否有团购 int 根据是否有团购来筛选返回的商户，1:有，0:没有

public String has\_online\_reservation;// 是否支持在线预订 int

// 根据是否支持在线预订来筛选返回的商户，1:有，0:没有

public String sort;// 结果排序 int

// 结果排序，1:默认，2:星级高优先，3:产品评价高优先，4:环境评价高优先，5:服务评价高优先，6:点评数量多优先，7:离传入经纬度坐标距离近优先，8:人均价格低优先，9：人均价格高优先

public String limit;// 结果条目数上限 int 每页返回的商户结果条目数上限，最小值1，最大值40，如不传入默认为20

public String page;// 页码 int 页码，如不传入默认为1，即第一页

}

}

package org.icscn.server.standalone.net.model.seller;

import java.util.ArrayList;

import java.util.List;

import org.icscn.server.standalone.net.model.CommonRepMessage;

public class SearchSjRepMessage extends CommonRepMessage {

/\*\*

\*

\*/

private static final long serialVersionUID = 1L;

public List<DataList> DATA\_LIST = new ArrayList<DataList>();

public static class DataList {

public String avg\_price;// 人均价格 int 人均价格，单位:元，若没有人均，返回-1

public String review\_count;// 点评数量 int 点评数量

public String review\_list\_url;// 点评页面URL链接 list 点评页面URL链接

public String distance;// 商户与参数坐标的距离 int

// 商户与参数坐标的距离，单位为米，如不传入经纬度坐标，结果为-1

public String business\_url;// 商户页面链接 string 商户页面链接

public String photo\_url;// 照片链接 string 照片链接，照片最大尺寸700×700

public String s\_photo\_url;// 小尺寸照片链接 string 小尺寸照片链接，照片最大尺寸278×200

public String photo\_count;// 照片数量 int 照片数量

public String photo\_list\_url;// 照片页面URL链接 list 照片页面URL链接

public String has\_coupon;// 是否有优惠券 int 是否有优惠券，0:没有，1:有

public String coupon\_id;// 优惠券ID int 优惠券ID

public String coupon\_description;// 优惠券描述 string 优惠券描述

public String coupon\_url;// 优惠券页面链接 string 优惠券页面链接

public String has\_deal;// 是否有团购 int 是否有团购，0:没有，1:有

public String deal\_count;// 商户当前在线团购数量 int 商户当前在线团购数量

public List<Deal> deals = new ArrayList<Deal>(); // 团购列表 list 团购列表

public String has\_online\_reservation;// 是否有在线预订 int 是否有在线预订，0:没有，1:有

public String online\_reservation\_url;// 在线预订页面链接 string

// 在线预订页面链接，目前仅返回HTML5站点链接

public String business\_id;// 商户ID int 商户ID

public String name;// 商户名 string 商户名

public String branch\_name;// 分店名 string 分店名

public String address;// 地址 string 地址

public String telephone;// 带区号的电话 string 带区号的电话

public String city;// 所在城市 string 所在城市

public String regions;// 所在区域信息列表 list 所在区域信息列表，如[徐汇区，徐家汇]

public String categories;// 所属分类信息列表 list 所属分类信息列表，如[宁波菜，婚宴酒店]

public String latitude;// 纬度坐标 float 纬度坐标

public String longitude;// 经度坐标 float 经度坐标

public String avg\_rating;// 星级评分 float 星级评分，5.0代表五星，4.5代表四星半，依此类推

public String rating\_img\_url;// 星级图片链接 string 星级图片链接

public String rating\_s\_img\_url;// 小尺寸星级图片链接 string 小尺寸星级图片链接

public String product\_grade;// 产品/食品口味评价 int

// 产品/食品口味评价，1:一般，2:尚可，3:好，4:很好，5:非常好

public String decoration\_grade;// 环境评价 int

// 环境评价，1:一般，2:尚可，3:好，4:很好，5:非常好

public String service\_grade;// 服务评价 int 服务评价，1:一般，2:尚可，3:好，4:很好，5:非常好

public String product\_score;// 产品/食品口味评价单项分 float

// 产品/食品口味评价单项分，精确到小数点后一位（十分制）

public String decoration\_score;// 环境评价单项分 float 环境评价单项分，精确到小数点后一位（十分制）

public String service\_score;// 服务评价单项分 float 服务评价单项分，精确到小数点后一位（十分制）

public String area\_no;// 区域编号 string 区域编号

public String build\_no;// 建筑编号 string 建筑编号

public String floor\_no;// 楼层编号 string 楼层编号

public String door\_no;// 房间编号 string 房间编号

public static class Deal {

public String id;// /团购ID string 团购ID

public String description;// 团购描述 string 团购描述

public String url;// 团购页面链接 string 团购页面链接

}

}

}

package org.icscn.server.standalone.net.model.seller;

import org.icscn.server.standalone.net.model.BaseObject;

public class SearchDealReqMessage extends BaseObject {

public Message MESSAGE = new Message();

public static class Message {

public String deal\_id;//

public String latitude;// 纬度坐标 float 纬度坐标，须与经度坐标同时传入，与城市名称二者必选其一传入

public String longitude;// 经度坐标 float 经度坐标，须与纬度坐标同时传入，与城市名称二者必选其一传入

public String radius;// 搜索半径 int 搜索半径，单位为米，最小值1，最大值5000，如不传入默认为1000

public String area\_no;// 区域编号 string 区域编号(经纬度没有传入的情况下必须传入)

public String build\_no;// 建筑编号 string 建筑编号

public String floor\_no;// 楼层编号 string 楼层编号

public String door\_no;// 房间编号 string 房间编号

public String region;// 城市区域名 string

// 城市区域名，可选范围见相关API返回结果（不含返回结果中包括的城市名称信息），如传入城市区域名，则城市名称必须传入

public String category;// 分类名 string

// 分类名，可选范围见相关API返回结果；支持同时输入多个分类，以逗号分隔，最大不超过5个。

public String keyword;// 关键词 string 关键词，搜索范围包括商户名、地址、标签等

public String is\_local;// 是否是本地单来筛选 int 根据是否是本地单来筛选返回的团购，1:是，0:不是

public String sort;// 结果排序 int

// 结果排序，1:默认，2:星级高优先，3:产品评价高优先，4:环境评价高优先，5:服务评价高优先，6:点评数量多优先，7:离传入经纬度坐标距离近优先，8:人均价格低优先，9：人均价格高优先

public String limit;// 结果条目数上限 int 每页返回的商户结果条目数上限，最小值1，最大值40，如不传入默认为20

public String page;// 页码 int 页码，如不传入默认为1，即第一页

}

}

package org.icscn.server.standalone.net.model.comm;

import java.util.ArrayList;

import java.util.List;

import org.icscn.server.standalone.net.model.BaseObject;

/\*\*

\*

\* @author felly

\*

\*/

public class InviAddGroupReqMessage extends BaseObject {

public Message MESSAGE = new Message();

public static class Message {

public List<Invi> INVI\_LIST = new ArrayList<Invi>();

public String USER\_NO;

}

public static class Invi {

public String GROUP\_ID, RCV\_NO, CONTENT\_MSG;

}

}

package org.icscn.server.standalone.net.model.comm;

import org.icscn.server.standalone.net.model.BaseObject;

public class ApplyAddGroupReqMessage extends BaseObject {

public Message MESSAGE = new Message();

public static class Message {

public String GROUP\_ID, USER\_NO, CONTENT\_MSG;

}

}

package org.icscn.server.standalone.net.dao.comm;

import java.sql.ResultSet;

import org.icscn.server.standalone.db.DaoUtil;

import org.icscn.server.standalone.net.model.comm.AddFriendReqMessage;

/\*\*

\*

\* @author felly

\*

\*/

public class CommDao {

/\*\*

\* 通过mapno查找区域，楼宇，楼层信息

\*/

public ResultSet getLoc(String mapNo) throws Exception {

String sql = "select `FLOOR\_NO` , `BUILD\_NO` , `AREA\_NO` from `d\_floor` where `MAP\_NO`=?";

ResultSet rs = DaoUtil.queryData(sql, new Object[] { mapNo });

return rs;

}

/\*\*

\* 查询地图编号

\*/

public String getMapNo(String mac) throws Exception {

String sql = "select `MAP\_ID` from `macadresslist` where `MAC`=? ";

ResultSet rs = DaoUtil.queryData(sql, new Object[] { mac });

if (rs.next()) {

return rs.getString("MAP\_ID");

}

return null;

}

/\*\*

\* 判断是不是已经是好友了

\*/

public boolean isFriend(String userNo, String friendNo) throws Exception {

String sql = "select 1 from t\_user\_friend where user\_no=? and FRIEND\_USER\_NO=? ";

return DaoUtil.queryData(sql, new Object[] { userNo, friendNo }).next();

}

/\*\*

\* 获取好友列表

\*/

public ResultSet queryFriendList(String userNo) throws Exception {

String sql =

"SELECT k\_user.user\_no,\n" +

" k\_user.user\_name,\n" +

" t\_user\_friend.friend\_status as STATUS,\n" +

" t\_online\_user.area\_no,\n" +

" t\_online\_user.mp\_no,\n" +

" t\_online\_user.x map\_x,\n" +

" t\_online\_user.y map\_y,\n" +

" t\_user\_friend.last\_hd,\n" +

" '' description,\n" +

" k\_user\_ext.USER\_IMG\n" +

"\n" +

" FROM t\_user\_friend\n" +

" LEFT JOIN k\_user\n" +

" ON t\_user\_friend.friend\_user\_no = k\_user.user\_no\n" +

" left join k\_user\_ext\n" +

"\n" +

" on k\_user\_ext.USER\_NO = k\_user.user\_no\n" +

" LEFT JOIN t\_online\_user\n" +

" ON k\_user.user\_no = t\_online\_user.user\_no\n" +

"\n" +

" WHERE t\_user\_friend.user\_no = ?";

return DaoUtil.queryData(sql, new Object[] { userNo });

}

/\*\*

\* 获取好友列表

\*/

public ResultSet queryFriendByName(String userName) throws Exception {

String sql = "SELECT k\_user.user\_no, k\_user.user\_name, t\_user\_friend.friend\_status status,\n"

+ "\t\t\t t\_online\_user.area\_no, t\_online\_user.mp\_no, t\_online\_user.x map\_x,\n"

+ "\t\t\t t\_online\_user.y map\_y, t\_user\_friend.last\_hd, '' description\n"

+ "\tFROM t\_user\_friend\n"

+ "\tLEFT JOIN k\_user\n"

+ "\t\tON t\_user\_friend.friend\_user\_no = k\_user.user\_no\n"

+ "\tLEFT JOIN t\_online\_user\n"

+ "\t\tON k\_user.user\_no = t\_online\_user.user\_no\n"

+ " WHERE t\_user\_friend.user\_name like CONCAT('%',?,'%')";

return DaoUtil.queryData(sql, new Object[] { userName });

}

/\*\*

\* 获取好友列表

\*/

public ResultSet queryFriendByMap(String mapNo) throws Exception {

String sql = "SELECT k\_user.user\_no, k\_user.user\_name, t\_user\_friend.friend\_status status,\n"

+ "\t\t\t t\_online\_user.area\_no, t\_online\_user.mp\_no, t\_online\_user.x map\_x,\n"

+ "\t\t\t t\_online\_user.y map\_y, t\_user\_friend.last\_hd, '' description\n"

+ "\tFROM t\_user\_friend\n"

+ "\tLEFT JOIN k\_user\n"

+ "\t\tON t\_user\_friend.friend\_user\_no = k\_user.user\_no\n"

+ "\tLEFT JOIN t\_online\_user\n"

+ "\t\tON k\_user.user\_no = t\_online\_user.user\_no\n"

+ " WHERE t\_online\_user.mp\_no=? ";

return DaoUtil.queryData(sql, new Object[] { mapNo });

}

/\*\*

\* 发起好友申请

\*/

public void addFriend(AddFriendReqMessage model) throws Exception {

String sql = "insert into T\_USER\_ADDINFO(SEND\_USER\_NO,RCVER\_NO,CONTENT\_MSG,REQ\_TIME,STATUS)"

+ " values(?,?,?,now(),?)";

DaoUtil.executeQuery(sql, new Object[] { model.MESSAGE.SEND\_USER\_NO,

model.MESSAGE.RCVER\_NO, model.MESSAGE.CONTENT\_MSG, "01"

});

}

/\*\*

\* 保存处理结果

\*/

public void rcvFriend(String reqNo, String rlt, String reason)

throws Exception {

String sql = "update T\_USER\_ADDINFO set RCV\_DATE=now(),STATUS=?,CONFU\_RESON=? where id=?";

DaoUtil.executeQuery(sql, new Object[] { rlt, reason, reqNo

});

}

/\*\*

\* 查询请求

\*/

public ResultSet queryAddFriendReq(String reqNo) throws Exception {

String sql = "select (select user\_name from "

+ " k\_user user where user.user\_no="

+ "RCVER\_NO limit 1) RCV\_NAME,CONFU\_RESON, SEND\_USER\_NO,RCVER\_NO,CONTENT\_MSG from T\_USER\_ADDINFO where id=? ";

return DaoUtil.queryData(sql, new Object[] { reqNo });

}

/\*\*

\*

\* @param userNo

\* @return

\* @throws Exception

\*/

public ResultSet queryAddAddFriendList(String userNo) throws Exception {

String sql = "select ID,SEND\_USER\_NO,RCVER\_NO,CONTENT\_MSG,REQ\_TIME,STATUS from "

+ " T\_USER\_ADDINFO where RCVER\_NO=? and STATUS in('01','02') ";

return DaoUtil.queryData(sql, new Object[] { userNo });

}

/\*\*

\* 解除关系并记录

\*/

public void delFreind(String sendUserNo, String delUserNo, String delReason)

throws Exception {

String sql = "delete from T\_USER\_FRIEND where USER\_NO=? and FRIEND\_USER\_NO=? ";

DaoUtil.executeQuery(sql, new Object[] { sendUserNo, delUserNo });

sql = "insert into T\_USER\_UNADD(SEND\_USER\_NO,DEL\_USER\_NO,DEL\_REASON,DEL\_TIME)"

+ " values(?,?,?,now())";

DaoUtil.executeQuery(sql, new Object[] { sendUserNo, delUserNo,

delReason });

}

/\*\*

\* 添加好友

\*/

public void addFreind(String sendUserNo, String rcvUserNo) throws Exception {

String sql = "delete from T\_USER\_FRIEND where USER\_NO=? and FRIEND\_USER\_NO=? ";

DaoUtil.executeQuery(sql, new Object[] { sendUserNo, rcvUserNo });

sql = "insert into T\_USER\_FRIEND(USER\_NO,FRIEND\_USER\_NO,ADD\_TIME)"

+ " values(?,?,now())";

DaoUtil.executeQuery(sql, new Object[] { sendUserNo, rcvUserNo });

}

/\*\*

\* 添加系统消息

\*/

public void addSysMsg(String userNo, String rcvNo, String msg)

throws Exception {

String sql = "insert into T\_MSG(USER\_NO,RCV\_NO,MSG,STATUS,SEND\_TIME)"

+ " values(?,?,?,'01',now())";

DaoUtil.executeQuery(sql, new Object[] { userNo, rcvNo, msg });

}

/\*\*

\* 查询用户

\*/

public boolean haveUser(String userNo) throws Exception {

String sql = "select 1 from k\_user where user\_no=? ";

return DaoUtil.queryData(sql, new Object[] { userNo }).next();

}

}

package org.icscn.server.standalone.net.busii.comm;

import org.icscn.server.standalone.json.GsonFactory;

import org.icscn.server.standalone.net.IHandler;

import org.icscn.server.standalone.net.Response;

import org.icscn.server.standalone.net.Response.Status;

import org.icscn.server.standalone.net.dao.comm.CommDao;

import org.icscn.server.standalone.net.model.CommonRepMessage;

import org.icscn.server.standalone.net.model.comm.AddFriendReqMessage;

import com.aiyc.framework.annotation.Requestable;

import com.aiyc.framework.utils.StringUtils;

import com.google.gson.JsonElement;

/\*\*

\*

\* @author felly

\*

\*/

@Requestable(serverName = "ADD\_FREND\_REQ")

public class AddFriendHandler implements IHandler {

private CommDao dao = new CommDao();

/\*\*

\*

\*/

public Response handle(JsonElement data) throws Exception {

AddFriendReqMessage model = GsonFactory.getGsonInstance().fromJson(

data, AddFriendReqMessage.class);

CommonRepMessage rep = new CommonRepMessage();

String sendNo=model.MESSAGE.SEND\_USER\_NO;

String rcvNo=model.MESSAGE.RCVER\_NO;

if(StringUtils.isNullOrBank(sendNo)){

rep.RTN\_FLAG = "0";

rep.RTN\_MSG = "发送人不能为空！";

return new Response(Status.failed, rep.RTN\_MSG, rep.toJson());

}

if(StringUtils.isNullOrBank(rcvNo)){

rep.RTN\_FLAG = "0";

rep.RTN\_MSG = "不存在这个用户哦";

return new Response(Status.failed, rep.RTN\_MSG, rep.toJson());

}

/////判断发送人是否合法用户

if(!dao.haveUser(sendNo)){

rep.RTN\_FLAG = "0";

rep.RTN\_MSG = "发送人不合法！";

return new Response(Status.failed, rep.RTN\_MSG, rep.toJson());

}

if(!dao.haveUser(rcvNo)){

rep.RTN\_FLAG = "0";

rep.RTN\_MSG = "好友不存在！";

return new Response(Status.failed, rep.RTN\_MSG, rep.toJson());

}

/\*\*

\* 判断是否已经是好友了

\*/

if(dao.isFriend(sendNo, rcvNo)){

rep.RTN\_FLAG = "0";

rep.RTN\_MSG = "你们已经是好友了！";

return new Response(Status.failed, rep.RTN\_MSG, rep.toJson());

}

dao.addFriend(model);

rep.RTN\_FLAG = "1";

rep.RTN\_MSG = "操作成功！";

return new Response(Status.ok, rep.RTN\_MSG, rep.toJson());

}

}

package org.icscn.server.standalone.net.busii.comm;

import org.icscn.server.standalone.json.GsonFactory;

import org.icscn.server.standalone.net.IHandler;

import org.icscn.server.standalone.net.Response;

import org.icscn.server.standalone.net.Response.Status;

import org.icscn.server.standalone.net.dao.comm.GroupDao;

import org.icscn.server.standalone.net.model.CommonRepMessage;

import org.icscn.server.standalone.net.model.comm.ApplyAddGroupReqMessage;

import com.aiyc.framework.annotation.Requestable;

import com.aiyc.framework.utils.StringUtils;

import com.google.gson.JsonElement;

/\*\*

\* 申请入群

\*

\* @author felly

\*

\*/

@Requestable(serverName = "APPLY\_GROUP\_REP")

public class ApplyAddGroupHandler implements IHandler {

private GroupDao dao = new GroupDao();

public Response handle(JsonElement data) throws Exception {

ApplyAddGroupReqMessage model = GsonFactory.getGsonInstance().fromJson(

data, ApplyAddGroupReqMessage.class);

CommonRepMessage rep = new CommonRepMessage();

if ("".equals(StringUtils.nullToEmpty(model.MESSAGE.USER\_NO))) {

rep.RTN\_FLAG = "0";

rep.RTN\_MSG = "申请人必须传入！";

return new Response(Status.failed, rep.RTN\_MSG, rep.toJson());

}

if ("".equals(StringUtils.nullToEmpty(model.MESSAGE.GROUP\_ID))) {

rep.RTN\_FLAG = "0";

rep.RTN\_MSG = "群组编号必须传入！";

return new Response(Status.failed, rep.RTN\_MSG, rep.toJson());

}

dao.applyAddGroup(model.MESSAGE.GROUP\_ID, model.MESSAGE.USER\_NO,

model.MESSAGE.CONTENT\_MSG);

rep.RTN\_FLAG = "1";

rep.RTN\_MSG = "申请成功！";

return new Response(Status.ok, rep.RTN\_MSG, rep.toJson());

}

}

package org.icscn.server.standalone.net.busii.comm;

import org.icscn.server.standalone.json.GsonFactory;

import org.icscn.server.standalone.net.IHandler;

import org.icscn.server.standalone.net.Response;

import org.icscn.server.standalone.net.Response.Status;

import org.icscn.server.standalone.net.dao.comm.GroupDao;

import org.icscn.server.standalone.net.model.comm.CreateGroupRepMessage;

import org.icscn.server.standalone.net.model.comm.CreateGroupReqMessage;

import com.aiyc.framework.annotation.Requestable;

import com.aiyc.framework.utils.StringUtils;

import com.google.gson.JsonElement;

/\*\*

\* 组群

\*

\* @author felly

\*

\*/

@Requestable(serverName = "CREATE\_GROUP\_REQ")

public class CreateGroupHandler implements IHandler {

private GroupDao dao = new GroupDao();

public Response handle(JsonElement data) throws Exception {

CreateGroupReqMessage model =GsonFactory.getGsonInstance().fromJson(

data, CreateGroupReqMessage.class);

CreateGroupRepMessage rep = new CreateGroupRepMessage();

if ("".equals(StringUtils.nullToEmpty(model.MESSAGE.GROUP\_NO))) {

rep.RTN\_FLAG = "0";

rep.RTN\_MSG = "组群人必须传入！";

return new Response(Status.failed, rep.RTN\_MSG, rep.toJson());

}

if ("".equals(StringUtils.nullToEmpty(model.MESSAGE.GROUP\_NAME))) {

rep.RTN\_FLAG = "0";

rep.RTN\_MSG = "群组名称必须传入！";

return new Response(Status.failed, rep.RTN\_MSG, rep.toJson());

}

String groupId = dao.createGroup(model);

rep.RTN\_FLAG = "1";

rep.RTN\_MSG = "操作成功！";

rep.GROUP\_ID = groupId;

return new Response(Status.ok, rep.RTN\_MSG, rep.toJson());

}

}

package org.icscn.server.standalone.net.busii.comm;

import java.sql.ResultSet;

import java.util.ArrayList;

import java.util.List;

import org.icscn.server.standalone.json.GsonFactory;

import org.icscn.server.standalone.net.IHandler;

import org.icscn.server.standalone.net.Response;

import org.icscn.server.standalone.net.Response.Status;

import org.icscn.server.standalone.net.dao.comm.CommDao;

import org.icscn.server.standalone.net.model.GetUserReqMessage;

import org.icscn.server.standalone.net.model.comm.GetAddFriendRepMessage;

import com.aiyc.framework.annotation.Requestable;

import com.google.gson.JsonElement;

/\*\*

\*

\* @author felly

\*

\*/

@Requestable(serverName = "GET\_ADDFREND\_REQ")

public class GetAddFriendHandler implements IHandler {

private CommDao dao = new CommDao();

/\*\*

\*

\*/

public Response handle(JsonElement data) throws Exception {

GetUserReqMessage model = GsonFactory.getGsonInstance().fromJson(data,

GetUserReqMessage.class);

ResultSet rs = dao.queryAddAddFriendList(model.MESSAGE.USER\_NO);

GetAddFriendRepMessage reps = new GetAddFriendRepMessage();

List<GetAddFriendRepMessage.AddList> list = new ArrayList<GetAddFriendRepMessage.AddList>();

while (rs.next()) {

GetAddFriendRepMessage.AddList rep = new GetAddFriendRepMessage.AddList();

rep.REQ\_NO = rs.getString("ID");

rep.SEND\_USER\_NO = rs.getString("SEND\_USER\_NO");

rep.RCVER\_NO = rs.getString("RCVER\_NO");

rep.CONTENT\_MSG = rs.getString("CONTENT\_MSG");

rep.REQ\_TIME = rs.getString("REQ\_TIME");

rep.STATUS = rs.getString("STATUS");

list.add(rep);

}

if (list.size() > 0) {

reps.ADDFRENDLIST=list;

reps.RTN\_FLAG = "1";

reps.RTN\_MSG = "操作成功！";

return new Response(Status.ok, reps.RTN\_MSG, reps.toJson());

} else {

reps.RTN\_FLAG = "0";

reps.RTN\_MSG = "无查询结果";

return new Response(Status.failed, reps.RTN\_MSG, reps.toJson());

}

}

}

package org.icscn.server.standalone.net.busii.comm;

import java.util.ArrayList;

import java.util.List;

import org.icscn.server.standalone.json.GsonFactory;

import org.icscn.server.standalone.net.IHandler;

import org.icscn.server.standalone.net.Response;

import org.icscn.server.standalone.net.Response.Status;

import org.icscn.server.standalone.net.dao.comm.GroupDao;

import org.icscn.server.standalone.net.model.GetUserReqMessage;

import org.icscn.server.standalone.net.model.comm.GetAddGroupRepMessage;

import com.aiyc.framework.annotation.Requestable;

import com.aiyc.framework.component.CachedRowSet;

import com.google.gson.JsonElement;

@Requestable(serverName = "GET\_ADDGROUP\_REQ")

public class GetAddGroupHandler implements IHandler {

private GroupDao dao = new GroupDao();

public Response handle(JsonElement data) throws Exception {

GetUserReqMessage model = GsonFactory.getGsonInstance().fromJson(data,

GetUserReqMessage.class);

GetAddGroupRepMessage rep = new GetAddGroupRepMessage();

List<GetAddGroupRepMessage.AddGroup> ADDGROUPLIST = new ArrayList<GetAddGroupRepMessage.AddGroup>();

CachedRowSet cs = dao.getAddGroup(model.MESSAGE.USER\_NO);

while (cs.next()) {

GetAddGroupRepMessage.AddGroup g = new GetAddGroupRepMessage.AddGroup();

g.REQ\_NO = cs.getString("ID");

g.GROUP\_ID = cs.getString("GROUP\_ID");

g.USER\_NO = cs.getString("send\_user\_no");

g.RCV\_NO = cs.getString("rcv\_no");

g.CONTENT\_MSG = cs.getString("content\_msg");

g.REQ\_TIME = cs.getString("req\_time");

g.STATUS = cs.getString("status");

g.ADD\_SORT = cs.getString("add\_sort");

ADDGROUPLIST.add(g);

}

if (cs.getRowCount() > 0) {

rep.ADDGROUPLIST = ADDGROUPLIST;

rep.RTN\_FLAG = "1";

rep.RTN\_MSG = "查询成功！";

return new Response(Status.ok, rep.RTN\_MSG, rep.toJson());

} else {

rep.RTN\_FLAG = "0";

rep.RTN\_MSG = "查询失败，无请求！";

return new Response(Status.failed, rep.RTN\_MSG, rep.toJson());

}

}

}

package org.icscn.server.standalone.net.busii.comm;

import java.sql.ResultSet;

import java.util.List;

import org.icscn.server.standalone.json.GsonFactory;

import org.icscn.server.standalone.net.IHandler;

import org.icscn.server.standalone.net.Response;

import org.icscn.server.standalone.net.Response.Status;

import org.icscn.server.standalone.net.dao.comm.CommDao;

import org.icscn.server.standalone.net.model.GetUserReqMessage;

import org.icscn.server.standalone.net.model.comm.GetFriendRepMessage;

import com.aiyc.framework.annotation.Requestable;

import com.aiyc.framework.utils.ResultToObject;

import com.google.gson.JsonElement;

/\*\*

\*

\* @author felly

\*

\*/

@Requestable(serverName = "GET\_FRIEND\_REQ")

public class GetFriendHandler implements IHandler {

private CommDao dao = new CommDao();

/\*\*

\*

\*/

public Response handle(JsonElement data) throws Exception {

GetUserReqMessage model = GsonFactory.getGsonInstance().fromJson(data,

GetUserReqMessage.class);

GetFriendRepMessage rep = new GetFriendRepMessage();

ResultSet rs = dao.queryFriendList(model.MESSAGE.USER\_NO);

List<GetFriendRepMessage.Friends> friends = ResultToObject

.resultToBase(GetFriendRepMessage.Friends.class, rs);

if (friends.size() > 0) {

rep.FRIEND\_LIST = friends;

rep.RTN\_FLAG = "1";

rep.RTN\_MSG = "查询成功！";

return new Response(Status.ok, rep.RTN\_MSG, rep.toJson());

} else {

rep.RTN\_FLAG = "0";

rep.RTN\_MSG = "查询失败，无好友！";

return new Response(Status.failed, rep.RTN\_MSG, rep.toJson());

}

}

}

package org.icscn.server.standalone.net.busii.comm;

import java.util.List;

import org.icscn.server.standalone.json.GsonFactory;

import org.icscn.server.standalone.net.IHandler;

import org.icscn.server.standalone.net.Response;

import org.icscn.server.standalone.net.Response.Status;

import org.icscn.server.standalone.net.dao.comm.GroupDao;

import org.icscn.server.standalone.net.model.GetUserReqMessage;

import org.icscn.server.standalone.net.model.comm.GetGroupRepMessage;

import com.aiyc.framework.annotation.Requestable;

import com.aiyc.framework.component.CachedRowSet;

import com.aiyc.framework.utils.ResultToObject;

import com.google.gson.JsonElement;

@Requestable(serverName = "GET\_GROUP\_REQ")

public class GetGroupHandler implements IHandler {

private GroupDao dao = new GroupDao();

/\*\*

\*

\*/

public Response handle(JsonElement data) throws Exception {

GetUserReqMessage model = GsonFactory.getGsonInstance().fromJson(data,

GetUserReqMessage.class);

String userNo = model.MESSAGE.USER\_NO;

com.aiyc.framework.component.CachedRowSet set = (CachedRowSet) dao

.queryGroup(userNo);

GetGroupRepMessage rep = new GetGroupRepMessage();

if (set.getRowCount() > 0) {

List<GetGroupRepMessage.Groups> gs = ResultToObject.resultToBase(

GetGroupRepMessage.Groups.class, set);

rep.GROUP\_LIST = gs;

rep.RTN\_FLAG = "1";

rep.RTN\_MSG = "操作成功！";

return new Response(Status.ok, rep.RTN\_MSG, rep.toJson());

} else {

rep.RTN\_FLAG = "0";

rep.RTN\_MSG = "无群组！";

return new Response(Status.failed, rep.RTN\_MSG, rep.toJson());

}

}

}

package org.icscn.server.standalone.net.busii.comm;

import org.icscn.server.standalone.json.GsonFactory;

import org.icscn.server.standalone.net.IHandler;

import org.icscn.server.standalone.net.Response;

import org.icscn.server.standalone.net.Response.Status;

import org.icscn.server.standalone.net.dao.comm.GroupDao;

import org.icscn.server.standalone.net.model.CommonRepMessage;

import org.icscn.server.standalone.net.model.comm.InviAddGroupReqMessage;

import com.aiyc.framework.annotation.Requestable;

import com.google.gson.JsonElement;

/\*\*

\* 邀请入群

\*

\* @author felly

\*

\*/

@Requestable(serverName = "INVIADD\_GROUP\_REQ")

public class InviAddGroupHandler implements IHandler {

private GroupDao dao = new GroupDao();

public Response handle(JsonElement data) throws Exception {

CommonRepMessage rep = new CommonRepMessage();

InviAddGroupReqMessage model = GsonFactory.getGsonInstance().fromJson(

data, InviAddGroupReqMessage.class);

if ("".equals(model.MESSAGE.USER\_NO) || model.MESSAGE.USER\_NO == null) {

rep.RTN\_FLAG = "0";

rep.RTN\_MSG = "邀请人必须传入！";

return new Response(Status.failed, rep.RTN\_MSG, rep.toJson());

}

if (model.MESSAGE.INVI\_LIST.size() == 0) {

rep.RTN\_FLAG = "0";

rep.RTN\_MSG = "邀请明细必须传入！";

return new Response(Status.failed, rep.RTN\_MSG, rep.toJson());

}

if (model.MESSAGE.INVI\_LIST.size() > 50) {

rep.RTN\_FLAG = "0";

rep.RTN\_MSG = "一次邀请数量不能超过50！";

return new Response(Status.failed, rep.RTN\_MSG, rep.toJson());

}

dao.inviAddGroup(model);

rep.RTN\_FLAG = "1";

rep.RTN\_MSG = "操作成功！";

return new Response(Status.ok, rep.RTN\_MSG, rep.toJson());

}

}

package org.icscn.server.standalone.net.busii.comm;

import org.icscn.server.standalone.json.GsonFactory;

import org.icscn.server.standalone.net.IHandler;

import org.icscn.server.standalone.net.Response;

import org.icscn.server.standalone.net.Response.Status;

import org.icscn.server.standalone.net.dao.comm.GroupDao;

import org.icscn.server.standalone.net.model.CommonRepMessage;

import org.icscn.server.standalone.net.model.comm.KickGroupUserReqMessage;

import com.aiyc.framework.annotation.Requestable;

import com.aiyc.framework.utils.StringUtils;

import com.google.gson.JsonElement;

/\*\*

\*

\* @author felly

\*

\*/

@Requestable(serverName = "KICK\_USER\_REQ")

public class KickGroupUserHandler implements IHandler {

private GroupDao dao = new GroupDao();

public Response handle(JsonElement data) throws Exception {

KickGroupUserReqMessage model = GsonFactory.getGsonInstance().fromJson(

data, KickGroupUserReqMessage.class);

CommonRepMessage rep = new CommonRepMessage();

if ("".equals(StringUtils.nullToEmpty(model.MESSAGE.GROUP\_ID))) {

rep.RTN\_FLAG = "0";

rep.RTN\_MSG = "群组必须传入！";

return new Response(Status.failed, rep.RTN\_MSG, rep.toJson());

}

if ("".equals(StringUtils.nullToEmpty(model.MESSAGE.USER\_NO))) {

rep.RTN\_FLAG = "0";

rep.RTN\_MSG = "操作人必须传入！";

return new Response(Status.failed, rep.RTN\_MSG, rep.toJson());

}

if ("".equals(StringUtils.nullToEmpty(model.MESSAGE.KICK\_NO))) {

rep.RTN\_FLAG = "0";

rep.RTN\_MSG = "被剔除人必须传入！";

return new Response(Status.failed, rep.RTN\_MSG, rep.toJson());

}

if (!dao.haveDel(model.MESSAGE.GROUP\_ID, model.MESSAGE.USER\_NO)) {

rep.RTN\_FLAG = "0";

rep.RTN\_MSG = "无剔除权限！";

return new Response(Status.failed, rep.RTN\_MSG, rep.toJson());

}

if (!dao.isInGroup(model.MESSAGE.GROUP\_ID, model.MESSAGE.KICK\_NO)) {

rep.RTN\_FLAG = "0";

rep.RTN\_MSG = "被剔除人不在本群内！";

return new Response(Status.failed, rep.RTN\_MSG, rep.toJson());

}

dao.kickGroupUser(model.MESSAGE.GROUP\_ID, model.MESSAGE.USER\_NO,

model.MESSAGE.KICK\_NO, model.MESSAGE.KICK\_REASON);

rep.RTN\_FLAG = "1";

rep.RTN\_MSG = "操作成功！";

return new Response(Status.ok, rep.RTN\_MSG, rep.toJson());

}

}

package org.icscn.server.standalone.net.busii.comm;

import java.sql.ResultSet;

import org.icscn.server.standalone.json.GsonFactory;

import org.icscn.server.standalone.net.IHandler;

import org.icscn.server.standalone.net.Response;

import org.icscn.server.standalone.net.Response.Status;

import org.icscn.server.standalone.net.dao.comm.CommDao;

import org.icscn.server.standalone.net.model.CommonRepMessage;

import org.icscn.server.standalone.net.model.comm.AddFriendReqMessage;

import org.icscn.server.standalone.net.model.comm.RcvFriendReqMessage;

import com.aiyc.framework.annotation.Requestable;

import com.aiyc.framework.utils.StringUtils;

import com.google.gson.JsonElement;

/\*\*

\*

\* @author felly

\*

\*

\*/

@Requestable(serverName = "RCV\_FREND\_REQ")

public class RcvFriendHandler implements IHandler {

private CommDao dao = new CommDao();

public Response handle(JsonElement data) throws Exception {

RcvFriendReqMessage model = GsonFactory.getGsonInstance().fromJson(

data, RcvFriendReqMessage.class);

CommonRepMessage rep = new CommonRepMessage();

String reqNo = model.MESSAGE.REQ\_NO;

String rlt = model.MESSAGE.CONFIRM\_RSLT;

if ("".equals(StringUtils.nullToEmpty(reqNo))) {

rep.RTN\_FLAG = "0";

rep.RTN\_MSG = "请求编号为空！";

return new Response(Status.failed, rep.RTN\_MSG, rep.toJson());

}

if ("".equals(StringUtils.nullToEmpty(rlt))) {

rep.RTN\_FLAG = "0";

rep.RTN\_MSG = "处理结果为空！";

return new Response(Status.failed, rep.RTN\_MSG, rep.toJson());

}

// ////处理结果保存

dao.rcvFriend(reqNo, rlt, model.MESSAGE.CONFU\_RESON);

// 02忽略，03同意，04，拒绝,05,同意并添加对方

ResultSet rs = dao.queryAddFriendReq(reqNo);

// //处理结果处理

if ("05".equals(rlt)) {

while (rs.next()) {

AddFriendReqMessage add = new AddFriendReqMessage();

add.MESSAGE.CONTENT\_MSG = "";

add.MESSAGE.RCVER\_NO = rs.getString("SEND\_USER\_NO");

add.MESSAGE.SEND\_USER\_NO = rs.getString("RCVER\_NO");

dao.addFriend(add);

}

}

if ("05".equals(rlt) || "03".equals(rlt)) {

rs.beforeFirst();

while (rs.next()) {

dao.addFreind(rs.getString("SEND\_USER\_NO"), rs

.getString("RCVER\_NO"));

dao.addSysMsg(rs.getString("SEND\_USER\_NO"), rs

.getString("RCVER\_NO"), "添加好友"

+ rs.getString("RCV\_NAME") + "成功！");

}

}

if ("04".equals(rlt)) {

rs.beforeFirst();

while (rs.next()) {

dao.addSysMsg(rs.getString("SEND\_USER\_NO"), rs

.getString("RCVER\_NO"), "用户" + rs.getString("RCV\_NAME")

+ "拒绝，拒绝原因:" + rs.getString("CONFU\_RESON"));

}

}

rep.RTN\_FLAG = "1";

rep.RTN\_MSG = "处理成功！";

return new Response(Status.ok, rep.RTN\_MSG, rep.toJson());

}

}

package org.icscn.server.standalone.net.busii.comm;

import java.sql.ResultSet;

import org.icscn.server.standalone.json.GsonFactory;

import org.icscn.server.standalone.net.IHandler;

import org.icscn.server.standalone.net.Response;

import org.icscn.server.standalone.net.Response.Status;

import org.icscn.server.standalone.net.dao.comm.CommDao;

import org.icscn.server.standalone.net.dao.comm.GroupDao;

import org.icscn.server.standalone.net.model.CommonRepMessage;

import org.icscn.server.standalone.net.model.comm.RcvFriendReqMessage;

import com.aiyc.framework.annotation.Requestable;

import com.aiyc.framework.utils.StringUtils;

import com.google.gson.JsonElement;

/\*\*

\* 接收入群请求

\*

\* @author felly

\*

\*/

@Requestable(serverName = "RCV\_GROUP\_REQ")

public class RcvGroupHandler implements IHandler {

private GroupDao dao = new GroupDao();

public Response handle(JsonElement data) throws Exception {

RcvFriendReqMessage model = GsonFactory.getGsonInstance().fromJson(

data, RcvFriendReqMessage.class);

CommonRepMessage rep = new CommonRepMessage();

String reqNo = model.MESSAGE.REQ\_NO;

String rlt = model.MESSAGE.CONFIRM\_RSLT;

if ("".equals(StringUtils.nullToEmpty(reqNo))) {

rep.RTN\_FLAG = "0";

rep.RTN\_MSG = "请求编号为空！";

return new Response(Status.failed, rep.RTN\_MSG, rep.toJson());

}

if ("".equals(StringUtils.nullToEmpty(rlt))) {

rep.RTN\_FLAG = "0";

rep.RTN\_MSG = "处理结果为空！";

return new Response(Status.failed, rep.RTN\_MSG, rep.toJson());

}

// ////处理结果保存

dao.rcvGroup(reqNo, rlt, model.MESSAGE.CONFU\_RESON);

// 02忽略，03同意，04，拒绝

ResultSet rs = dao.queryAddGroupReq(reqNo);

if ("03".equals(rlt)) {

rs.beforeFirst();

while (rs.next()) {

dao.addGroupUser(rs.getString("group\_id"), rs

.getString("RCV\_NO"));

String sort=rs.getString("add\_sort");

String msg="";

if("01".equals(sort)){

msg="好友"

+ rs.getString("RCV\_NAME") + "同意加入群组"+rs.getString("group\_name");

}

else{

msg="同意加入群组"+rs.getString("group\_name");

}

(new CommDao()).addSysMsg(rs.getString("SEND\_USER\_NO"), rs

.getString("RCV\_NO"),msg );

}

}

if ("04".equals(rlt)) {

rs.beforeFirst();

while (rs.next()) {

String sort=rs.getString("add\_sort");

String msg="";

if("01".equals(sort)){

msg="好友"

+ rs.getString("RCV\_NAME") + "拒绝加入群组"+rs.getString("group\_name");

}

else{

msg="拒绝加入群组"+rs.getString("group\_name");

}

(new CommDao()).addSysMsg(rs.getString("SEND\_USER\_NO"), rs

.getString("RCV\_NO"),msg );

}

}

rep.RTN\_FLAG = "1";

rep.RTN\_MSG = "处理成功！";

return new Response(Status.ok, rep.RTN\_MSG, rep.toJson());

}

}

package org.icscn.server.standalone.net.busii.comm;

import java.util.List;

import org.icscn.server.standalone.json.GsonFactory;

import org.icscn.server.standalone.net.IHandler;

import org.icscn.server.standalone.net.Response;

import org.icscn.server.standalone.net.Response.Status;

import org.icscn.server.standalone.net.dao.comm.CommDao;

import org.icscn.server.standalone.net.model.comm.GetFriendRepMessage;

import org.icscn.server.standalone.net.model.comm.SearchUserReqMessage;

import com.aiyc.framework.annotation.Requestable;

import com.aiyc.framework.component.CachedRowSet;

import com.aiyc.framework.utils.ResultToObject;

import com.aiyc.framework.utils.StringUtils;

import com.google.gson.JsonElement;

@Requestable(serverName = "SEARCH\_USER\_REQ")

public class SearchFriendHandler implements IHandler {

private CommDao dao = new CommDao();

public Response handle(JsonElement data) throws Exception {

SearchUserReqMessage model = GsonFactory.getGsonInstance().fromJson(

data, SearchUserReqMessage.class);

GetFriendRepMessage rep = new GetFriendRepMessage();

CachedRowSet as = null, bs = null, cs = null;

if ("".equals(StringUtils.nullToEmpty(model.MESSAGE.USER\_NO))

&& "".equals(StringUtils.nullToEmpty(model.MESSAGE.USER\_NAME))

&& "".equals(StringUtils.nullToEmpty(model.MESSAGE.MP\_NO))

) {

rep.RTN\_FLAG = "0";

rep.RTN\_MSG = "必须输入查询条件！";

return new Response(Status.failed, rep.RTN\_MSG, rep.toJson());

}

if (!"".equals(StringUtils.nullToEmpty(model.MESSAGE.USER\_NO)))

as = (CachedRowSet) dao.queryFriendList(model.MESSAGE.USER\_NO);

if (!"".equals(StringUtils.nullToEmpty(model.MESSAGE.USER\_NAME))) {

bs = (CachedRowSet) dao.queryFriendByName(model.MESSAGE.USER\_NAME);

}

if (!"".equals(StringUtils.nullToEmpty(model.MESSAGE.MP\_NO))) {

cs = (CachedRowSet) dao.queryFriendByMap(model.MESSAGE.MP\_NO);

}

CachedRowSet rs = null;

int a = 0, b = 0, c = 0;

if (as != null) {

rs = as;

a = 1;

} else if (bs != null) {

rs = bs;

b = 1;

} else if (cs != null) {

rs = cs;

c = 1;

}

if (rs == null) {

rep.RTN\_FLAG = "0";

rep.RTN\_MSG = "无查询结果！";

return new Response(Status.failed, rep.RTN\_MSG, rep.toJson());

}

if (a == 1 && b == 1 && c == 1) {

rs.union(bs);

rs.union(cs);

} else if (a == 0 && b == 1 && c == 1) {

rs.union(cs);

} else if (a == 1 && b == 0 && c == 1) {

rs.union(cs);

} else if (a == 1 && b == 1 && c == 0) {

rs.union(bs);

}

List<GetFriendRepMessage.Friends> friends = ResultToObject

.resultToBase(GetFriendRepMessage.Friends.class, rs);

if (friends.size() > 0) {

rep.FRIEND\_LIST = friends;

rep.RTN\_FLAG = "1";

rep.RTN\_MSG = "查询成功！";

return new Response(Status.ok, rep.RTN\_MSG, rep.toJson());

} else {

rep.RTN\_FLAG = "0";

rep.RTN\_MSG = "查询失败，无结果！";

return new Response(Status.failed, rep.RTN\_MSG, rep.toJson());

}

}

}

package org.icscn.server.standalone.net.busii.comm;

import java.util.List;

import org.icscn.server.standalone.json.GsonFactory;

import org.icscn.server.standalone.net.IHandler;

import org.icscn.server.standalone.net.Response;

import org.icscn.server.standalone.net.Response.Status;

import org.icscn.server.standalone.net.dao.comm.GroupDao;

import org.icscn.server.standalone.net.model.comm.GetGroupRepMessage;

import org.icscn.server.standalone.net.model.comm.SearchGroupReqMessage;

import com.aiyc.framework.annotation.Requestable;

import com.aiyc.framework.component.CachedRowSet;

import com.aiyc.framework.utils.ResultToObject;

import com.google.gson.JsonElement;

/\*\*

\* 群组搜索

\*

\* @author felly

\*

\*/

@Requestable(serverName = "SEARCH\_GROUP\_REQ")

public class SearchGroupHandler implements IHandler {

private GroupDao dao = new GroupDao();

public Response handle(JsonElement data) throws Exception {

SearchGroupReqMessage model = GsonFactory.getGsonInstance().fromJson(

data, SearchGroupReqMessage.class);

com.aiyc.framework.component.CachedRowSet set = (CachedRowSet) dao

.searchGroup(model.MESSAGE.GROUP\_ID, model.MESSAGE.GROUP\_NAME,

model.MESSAGE.GROUP\_FUN);

GetGroupRepMessage rep = new GetGroupRepMessage();

if (set != null && set.getRowCount() > 0) {

List<GetGroupRepMessage.Groups> gs = ResultToObject.resultToBase(

GetGroupRepMessage.Groups.class, set);

rep.GROUP\_LIST = gs;

rep.RTN\_FLAG = "1";

rep.RTN\_MSG = "操作成功！";

return new Response(Status.ok, rep.RTN\_MSG, rep.toJson());

} else {

rep.RTN\_FLAG = "0";

rep.RTN\_MSG = "无群组！";

return new Response(Status.failed, rep.RTN\_MSG, rep.toJson());

}

}

}

package org.icscn.server.standalone.net.busii.comm;

import org.icscn.server.standalone.json.GsonFactory;

import org.icscn.server.standalone.net.IHandler;

import org.icscn.server.standalone.net.Response;

import org.icscn.server.standalone.net.Response.Status;

import org.icscn.server.standalone.net.dao.comm.GroupDao;

import org.icscn.server.standalone.net.model.CommonRepMessage;

import org.icscn.server.standalone.net.model.comm.KickGroupUserReqMessage;

import com.aiyc.framework.annotation.Requestable;

import com.aiyc.framework.utils.StringUtils;

import com.google.gson.JsonElement;

/\*\*

\* 退出群组

\* @author felly

\*/

@Requestable(serverName = "UN\_GROUP\_REQ")

public class UnGroupHandler implements IHandler {

private GroupDao dao = new GroupDao();

public Response handle(JsonElement data) throws Exception {

KickGroupUserReqMessage model = GsonFactory.getGsonInstance().fromJson(

data, KickGroupUserReqMessage.class);

CommonRepMessage rep = new CommonRepMessage();

if ("".equals(StringUtils.nullToEmpty(model.MESSAGE.GROUP\_ID))) {

rep.RTN\_FLAG = "0";

rep.RTN\_MSG = "群组必须传入！";

return new Response(Status.failed, rep.RTN\_MSG, rep.toJson());

}

if ("".equals(StringUtils.nullToEmpty(model.MESSAGE.USER\_NO))) {

rep.RTN\_FLAG = "0";

rep.RTN\_MSG = "操作人必须传入！";

return new Response(Status.failed, rep.RTN\_MSG, rep.toJson());

}

if (!dao.isInGroup(model.MESSAGE.GROUP\_ID, model.MESSAGE.KICK\_NO)) {

rep.RTN\_FLAG = "0";

rep.RTN\_MSG = "被剔除人不在本群内！";

return new Response(Status.failed, rep.RTN\_MSG, rep.toJson());

}

dao.unGroupUser(model.MESSAGE.GROUP\_ID, model.MESSAGE.USER\_NO,

model.MESSAGE.KICK\_REASON);

rep.RTN\_FLAG = "1";

rep.RTN\_MSG = "操作成功！";

return new Response(Status.ok, rep.RTN\_MSG, rep.toJson());

}

}