1：操作：点击-》食品类

2：本页面地址：

跳转到的页面地址：

3：涉及到的表：

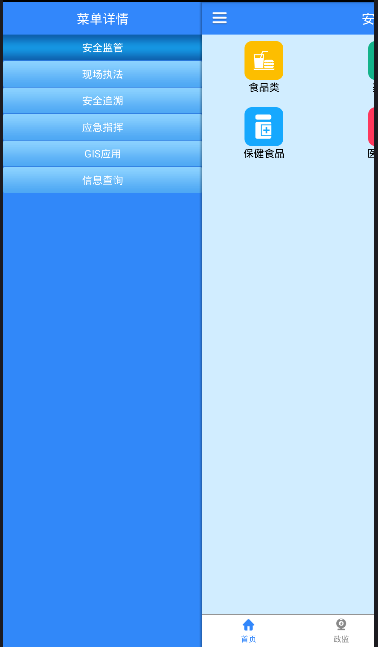
4：功能实现思路和数据流描述

5：代码

6：涉及到的流程图

7：跳转到的页面

**一：安全监管-》食品类**



1：操作：点击-》食品类

2：页面地址：

**package** com.askj.tangyin.drug.secondpager;

**public** **class** SafetyRegulationPager **extends** BaseMenuDetailPager **implements**

OnItemClickListener {

3：涉及到的表：aa10

4：功能实现思路和数据流描述

食品类aaa101=SPSC，aaa100=COMDALEI去数据

5：代码

（1）**case** 0:

//食品类

FieldContance.*itemid* = "2016051018315285615607502";

itemid = "2016051018315285615607502";

category = "spsc";

intent = **new** Intent(mActivity,EnterpriseCategoryActivity.**class**);

sp.edit().putString("itemid",itemid).commit();

sp.edit().putString("category",category).commit();

intent.putExtra("itemid", itemid);

mActivity.startActivity(intent);

**break**;

**case** 1:

（2） **final** String path = Contances.*QueryEnterpriseUrl*

+ "?aaa100=comdalei" + "&userid=" + userid+ "&aaa101=" + category;

// 查询所有企业接口

**public** **static** String *QueryEnterpriseUrl* = *baseUrl* + "getAa10Byaaa101toJSON";

（3）

**public** List getAa10ListByaaa101(Aa10DTO dto) **throws** Exception {

// sql查询语句

StringBuffer sb = **new** StringBuffer();

sb.append(" select a.aaa102 comdalei , a.\* ");

**if**(dto.getUserid()!=**null** && !"".equals(dto.getUserid())){

sb.append(" , (select b.ORGID from sysuser b where b.USERID= '")

.append(dto.getUserid()).append( "') ORGID ");

sb.append(" , (select c.AAA027 from sysuser c where c.USERID= '")

.append(dto.getUserid()).append( "') AAA027 ");

}

sb.append(" from Aa10 a WHERE a.AAA100 = '")

.append(dto.getAaa100().toUpperCase()).append("'");

sb.append(" and a.AAA101 = '").append(dto.getAaa101().toUpperCase())

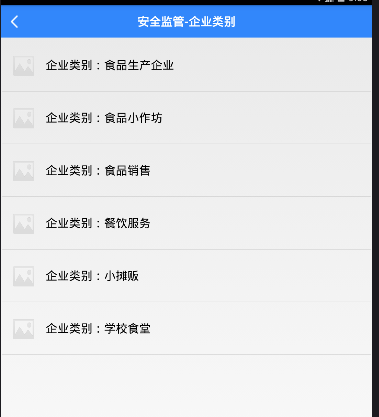
.append("'");

sb.append(" order by a.AAA105 , a.AAA102,a.aaa104 asc ");

String sql = sb.toString();

6：涉及到的流程图

7：跳转到的页面



1：操作：点击-》企业类型，食品生产企业

2：页面地址：

**package** com.askj.tangyin.drug.app;

**public** **class** EnterpriseTableActivity **extends** Activity **implements** OnClickListener, OnItemClickListener {

3：涉及到的表： pcompany,pcompanycomdalei,bscheckmaster, bscheckplan，aa10, ombasetype,omcheckgroup，bscheckmaster , bscheckplan

4：功能实现思路和数据流描述

5：代码

（1） @Override

**public** **void** onItemClick(AdapterView<?> parent, View view,

**int** position, **long** id) {

aaz093 = categoryBeanInfo.get(position).aaz093;//企业大类对应的表id

sp.edit().putString("aaz093", aaz093).commit();

Intent intent = **new** Intent(ctx,

EnterpriseTableActivity.**class**);

startActivity(intent);

}

（2） **final** String path = Contances.*QueryAllEnterpriseUrl* + "?userid=" + userId + "&aaz093=" + aaz093

+ "&page=" + page;

// 查询所有企业接口

**public** **static** String *QueryAllEnterpriseUrl* = *baseUrl* + "queryCompanyList";

（3）

//当前年份时间

dto.setYear(String.*valueOf*(DateUtil.*convertDateToYear*(**new** Date())));

**if**(dto.getItemid()!= **null** && !"".equals(dto.getItemid())){

StringBuffer sb = **new** StringBuffer();

sb.append(" select (select AAZ093 from aa10 where AAA100='comdalei' and AAA102=e.AAA102 ) as aaz093, a.comxkfw, a.comid,a.comzzzmbh,a.comzzjgdm ,a.comdm ,a.comxiaolei, cd.comdalei,");

sb.append(" a.comgddh,(select count(\*) from bscheckmaster f , bscheckplan g where f.planid=g.planid and f.operatedate like :year");

sb.append(" and f.resultstate =4 and f.comid=a.comid) as resultcount, ");

sb.append(" a.commc ,a.comfrhyz ,a.comdz,a.comfrsfzh,a.comyddh,a.aaa027,a.orgid,a.comzzzm ,a.comxkzbh

from pcompany a ,pcompanycomdalei cd, ");

sb.append(" (select DISTINCT b.AAA102 from aa10 b, ombasetype o,omcheckgroup c where AAA100 ='COMDALEI' and b.AAZ093 = o.basetype ");

sb.append(" and c.itemid=o.itemtype and c.itempid= :itemid and b.AAZ093=:aaz093 and 1=1");

sb.append(" ) e where cd.comdalei = e.AAA102 AND a.comid = cd.comid and 1=1 and a.commc like :commc");

//gu20161209 sb.append(" order by resultcount , a.comid desc ");

sb.append(" order by resultcount desc ");

sql = sb.toString();

String[] ParaName = **new** String[] { "year","itemid", "aaz093","commc"};

**int**[] ParaType = **new** **int**[] { Types.*VARCHAR*, Types.*VARCHAR*, Types.*VARCHAR*,Types.*VARCHAR*};

sql = QueryFactory.*getSQL*(sql, ParaName, ParaType, dto);

System.*out*.println("sqlsqlsqlsql itemid " + sql);

} **else** {//食品类

// 是否启用安全监管任务分派功能

Aa01 aa01 = SysmanageUtil.*getAa01*("SFQYAQJGRWFP");

StringBuffer sb = **new** StringBuffer();

sb.append(" select a.comzzzmbh,a.comxkfw,a.comzzjgdm ,a.comdz,a.comid,a.comdm ,a.commc ,a.orgid , a.comfrhyz ,cd.comdalei,");

sb.append(" a.comgddh,(select count(\*) from bscheckmaster f , bscheckplan g where f.planid=g.planid and f.operatedate like :year");

sb.append(" and f.resultstate =4 and f.comid=a.comid) as resultcount, ");

sb.append(" a.comfrsfzh,a.comyddh,a.aaa027,a.comzzzm,a.comxiaolei,a.comxkzbh ");

sb.append(" from pcompany a, aa10 b, pcompanycomdalei cd ");

**if** (aa01.getAaa005() != **null** && "1".equals(aa01.getAaa005())) {

sb.append(" , bschecktaskdetail detail, bschecktaskperson per ");

}

sb.append(" where cd.comid = a.comid ");

sb.append(" and cd.comdalei = b.AAA102 ");

sb.append(" and b.aaa100='comdalei' ");

sb.append(" and b.AAZ093 =:aaz093 ");

sb.append(" and a.commc like :commc");

**if** (aa01.getAaa005() != **null** && "1".equals(aa01.getAaa005())) {

sb.append(" and a.comid = detail.comid and per.userid =:userid GROUP BY a.comid ");

}

//gu20161209 sb.append(" order by resultcount, a.comid desc ");

sb.append(" order by resultcount desc ");

sql = sb.toString();

String[] ParaName = **new** String[] { "year", "aaz093", "commc", "userid"};

**int**[] ParaType = **new** **int**[] { Types.*VARCHAR*, Types.*VARCHAR*,Types.*VARCHAR*, Types.*VARCHAR* };

sql = QueryFactory.*getSQL*(sql, ParaName, ParaType, dto);

System.*out*.println("sqlsqlsqlsql aaz093" + sql);

}

6：涉及到的流程图

7：跳转到的页面



1：操作：点击-》第一个企业

2：本页面地址：

**package** com.askj.tangyin.drug.app;

**public** **class** EnterpriseTableActivity **extends** Activity **implements** OnClickListener, OnItemClickListener {

跳转到的页面地址：

**package** com.askj.tangyin.drug.app;

**public** **class** EnterpriseDetailActivity **extends** Activity **implements** OnClickListener, OnItemClickListener {

3：涉及到的表：看下面代码，sql可以优化

4：功能实现思路和数据流描述

5：代码

（1）// **TODO** listView的点击事件

@Override

**public** **void** onItemClick(AdapterView<?> parent, View view, **int** position, **long** id) {

String comid = companyInfo.get(position).comid;

String comdalei = companyInfo.get(position).comdalei;

String enTel = companyInfo.get(position).comyddh;

String itemid = sp.getString("itemid", "");

/\*\*

\*

\* 根据企业大类判断是否可以跳转到计划界面

\*/

**if** ("2016051018315285615607502".equals(itemid)) {

sp.edit().putString("comid", comid).commit();

sp.edit().putString("comdalei", comdalei).commit();

sp.edit().putString("comenTel", enTel).commit();

sp.edit().putString("companyCode", companyInfo.get(position).comfrhyz).commit();

Intent intent = **new** Intent(ctx, EnterpriseDetailActivity.**class**);

intent.putExtra("CompanyName", companyInfo.get(position).commc);

intent.putExtra("CompanyCode", companyInfo.get(position).comfrhyz);

startActivity(intent);

}

}

（2）

**final** String aaz093 = sp.getString("aaz093", "");

// **TODO** 联网操作

**new** Thread() {

**public** **void** run() {

**try** {

String comid = sp.getString("comid", "");

String itemid = sp.getString("itemid", "");

String userid = sp.getString("userid", "");

// **TODO** 获取结果ID接口

**final** String path = Contances.*QueryPlanTableUrl* + "?itemid=" + itemid + "&comid=" + comid

+ "&aaz093=" + aaz093 + "&userid=" + userid;

（3）

// 根据企业类型查询计划信息

**public** **static** String *QueryPlanTableUrl* = *baseUrl* + "getPlansByqyType";

（4）

**public** Map queryPlansByqyType(HttpServletRequest request, BscheckplanDTO dto)

**throws** Exception {

// 获取用户id

String userid = StringHelper.*showNull2Empty*(request.getParameter("userid"));

**if** ("".equals(userid)) {

**throw** **new** BusinessException("用户ID不能为空！");

}

// 使用字符串缓冲器类拼接查询语句

StringBuffer sb = **new** StringBuffer();

// 是否启用安全监管任务分派功能

Aa01 aa01 = SysmanageUtil.*getAa01*("SFQYAQJGRWFP");

sb.append(" select a.\*,e.itemid from bscheckplan a, ");

**if** (aa01.getAaa005() != **null** && "1".equals(aa01.getAaa005())) {

sb.append(" bschecktask b, bschecktaskperson per, ");

}

sb.append(" ( select DISTINCT b.planid ,c.itempid as itemid from bscheckpicset b ,omcheckgroup c ");

sb.append(" where b.itemid = c.itemid ");

sb.append(" and c.itempid in( select o.itemtype as itemid from aa10 b, ombasetype o,omcheckgroup c ");

sb.append(" where AAA100 ='COMDALEI' and b.AAZ093 = o.basetype and c.itemid=o.itemtype and ");

sb.append(" c.itempid =:itemid and b.AAZ093=:aaz093 and 1=1 order by b.AAA102 desc ");

sb.append(" ) and 1=1 ) e ");

sb.append(" where a.planid =e.planid and find\_in\_set(:aaz093,a.plantypearea) ");

**if** (aa01.getAaa005() != **null** && "1".equals(aa01.getAaa005())) {

sb.append(" AND a.planid = b.planid AND b.taskid = per.taskid AND per.userid = '").append(userid).append("' ");

}

sb.append(" order by a.planid desc ");

String sql = sb.toString();

String[] paramName = **new** String[] { "comid", "itemid","aaz093" ,"aaz093" };

6：涉及到的流程图

7：跳转到的页面



1：操作：点击-》食品类

2：本页面地址：

**package** com.askj.tangyin.drug.app;

**public** **class** EnterpriseDetailActivity **extends** Activity **implements** OnClickListener, OnItemClickListener {

跳转到的页面地址：

**package** com.askj.tangyin.drug.app;

**public** **class** PlanCheckHistoryActivity **extends** Activity **implements** OnItemClickListener, OnClickListener {

3：涉及到的表：

4：功能实现思路和数据流描述

5：代码

（1）

@Override

**public** **void** onItemClick(AdapterView<?> parent, View view, **int** position, **long** id) {

// Intent intent = new Intent(ctx, DailySupervisionActivity.class);

Intent intent = **new** Intent(ctx, PlanCheckHistoryActivity.**class**);

intent.putExtra("planid", planTables.get(position).planid);

intent.putExtra("itemid", planTables.get(position).itemid);

startActivity(intent);

}

（2）

**private** **void** initData() {

userName = sp.getString("userName", "");

Intent intent = getIntent();

planid = intent.getStringExtra("planid");

itemid = intent.getStringExtra("itemid");

ProgressDlgUtil.*showProgressDlg*("正在拼命加载中，请稍后...", ctx);

**new** Thread() {

**public** **void** run() {

**try** {

String comid = sp.getString("comid", "");

// **TODO** 获取结果ID接口

**final** String path = Contances.*GETOLDPLANLIST* + "?comid=" + comid + "&planid=" + planid;

// 安全监管获取计划历史记录

**public** **static** String *GETOLDPLANLIST* = *baseUrl* + "getoldPlanList";

（3）

/\*\*

\* 查询历史记录，(手机端service方法)

\*

\* **@param** dto企业实体类

\* **@param** pd分页

\* **@return**

\* **@throws** Exception

\*/

**public** Map getoldPlanList(HttpServletRequest request, BsCheckMasterDto dto)

**throws** Exception {

dto.setYear(String.*valueOf*(DateUtil.*convertDateToYear*(**new** Date())));

// 使用字符串缓冲器类拼接查询语句

StringBuffer sb = **new** StringBuffer();

sb.append(" select b.plantitle , a.resultid, a.operatedate ,a.resultdate , a.resultstate, a.resultremark , ");

sb.append("(select CONCAT(c.username,'(',c.DESCRIPTION,')') from sysuser c where c.userid=a.operateperson ) as operateperson ");

sb.append(" from bscheckmaster a , bscheckplan b ");

sb.append("where a.planid=b.planid ");//and a.resultstate <> 1

sb.append(" and a.operatedate like :year ");

sb.append("and a.comid =:comid ");

sb.append(" and a.planid=:planid ");

sb.append(" and a.qtbwid=:qtbwid ");

sb.append("order by a.operatedate desc,a.resultstate");

String sql = sb.toString();

6：涉及到的流程图

7：跳转到的页面



1：操作：点击-》新增检查

2：本页面地址：

**package** com.askj.tangyin.drug.app;

**public** **class** PlanCheckHistoryActivity **extends** Activity **implements** OnItemClickListener, OnClickListener {

跳转到的页面地址：

未检查完

**package** com.askj.tangyin.drug.app;

**public** **class** DailySupervisionActivity **extends** Activity **implements** OnClickListener {

已检查完

**package** com.askj.tangyin.drug.app;

**public** **class** ShowResultActivity **extends** Activity **implements** OnClickListener {

3：涉及到的表：

4：功能实现思路和数据流描述

5：代码

（1）

/\*\*

\* listview的条目点击事件

\*/

@Override

**public** **void** onItemClick(AdapterView<?> parent, View view, **int** position, **long** id) {

String resultstate = planCheckHistory.get(position).resultstate;

String resultid = planCheckHistory.get(position).resultid;

System.*out*.println(resultid + "======================resultid");

**if** (resultstate.equals("0")) {

Intent intent = **new** Intent(ctx, DailySupervisionActivity.**class**);

intent.putExtra("planid", planid);

intent.putExtra("resultid", resultid);

intent.putExtra("resultstate", resultstate);

intent.putExtra("itemid", itemid);

startActivity(intent);

finish();

} **else** **if** (resultstate.equals("4")) {

// **TODO** 检查完成就直接进打印预览界面

Intent intent = **new** Intent(ctx, ShowResultActivity.**class**);

intent.putExtra("resultid", resultid);

startActivity(intent);

finish();

} **else** {

// **TODO** 检查未完成

Intent intent = **new** Intent(ctx, DailySupervisionActivity.**class**);

intent.putExtra("planid", planid);

intent.putExtra("resultid", resultid);

intent.putExtra("resultstate", resultstate);

intent.putExtra("itemid", itemid);

startActivity(intent);

finish();

}

}

（2）

DailySupervisionActivity

**private** **void** initData() {

Intent intent = getIntent();

planid = intent.getStringExtra("planid");

itemid = intent.getStringExtra("itemid");、//检查项目类型，例如食品类

resultid = intent.getStringExtra("resultid");//未检查完成的有

String resultstate = intent.getStringExtra("resultstate") + "";

**if** (resultstate.equals("1") || resultstate.equals("2") || resultstate.equals("3")) {

bt\_all\_select\_match.setVisibility(View.*GONE*);

} **else** {

bt\_all\_select\_match.setVisibility(View.*VISIBLE*);

}

comid = sp.getString("comid", "");

aaa383 = sp.getString("aaa383", "");

aaa027 = sp.getString("aaa027", "");

String userName = sp.getString("userName", "");

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

tv\_check\_applicant.setText(userName);

}

// final String itemid = sp.getString("itemid", "");

System.*out*.println(planid + "=======" + itemid + "======" + comid);

// **TODO** 联网操作

**new** Thread() {

**public** **void** run() {

**try** {

// **TODO** 获取结果ID接口

**final** String path = Contances.*DailySupervisionUrl* + "?itemid=" + itemid + "&planid=" + planid

+ "&comid=" + comid + "&resultid=" + resultid;

System.*out*.println(comid);

URL url = **new** URL(path);

// 根据企业进行日常监管---现场核查

**public** **static** String *DailySupervisionUrl* = *baseUrl* + "getPlansByqyTypeAndid";

（3）

/\*\*

\* 查询出所有设置的明细为检查完的显示检查上次检查信息根据企业类别和计划id，(手机端service方法)

\*

\* **@param** dto企业实体类

\* **@param** pd分页

\* **@return**

\* **@throws** Exception

\*/

**public** Map queryPlansByqyTypeAndid(HttpServletRequest request,

BscheckplanDTO dto) **throws** Exception {

**int** planchecnnum=getplanChecknum(dto);//家检查总数

**int** resultchecnnum=getresultChecknum(dto);//实际检查总数

// 使用字符串缓冲器类拼接查询语句

StringBuffer sb = **new** StringBuffer();

sb.append(" select g.\*,IFNULL(h.detaildecide,'') as detaildecide,IFNULL(h.detailscore,0) as detailscore from (select c.itemid ,c.itemname, c.itemsortid,");

sb.append(" d.contentid,d.content,d.contentsortid,d.contentscore ,d.contentcode ,d.contentimpt from omcheckgroup c ,");

sb.append(" ( select a.\* from omcheckcontent a ,bscheckpicset b where a.contentid = b.contentid ");

sb.append(" and b.planid= :planid and 1=1) d where c.itemid = d.itemid ");

sb.append(" and c.itempid = :itemid and 1=1 ) g left join ( ");

**if**(dto.getResultid()!=**null** &&!"".equals(dto.getResultid())){

sb.append(" select DISTINCT f.contentid , f.detaildecide,f.detailscore from bscheckmaster e ,bscheckdetail f ");

sb.append("where e.resultid = f.resultid and e.resultstate<>4 and e.resultid=:resultid and e.planid= :planid and e.comid= :comid and 1=1 ");

}**else** {

//查询结果记录中最晚时间结束的结果项

**if**(resultchecnnum<planchecnnum && resultchecnnum!=0){//已经完成中，最后的结果不是检查完

sb.append(" select DISTINCT f.contentid , f.detaildecide,f.detailscore from bscheckdetail f , ");

sb.append(" ( select l.resultid from bscheckmaster l ,( select MAX(k.resultdate) as resultdate ,k.resultid from bscheckmaster k where ");

sb.append(" k.planid=:planid and k.comid=:comid and 1=1");

sb.append(" ) p where l.resultdate = p.resultdate ) j where f.resultid=j.resultid ");

}**else** {

sb.append(" select DISTINCT f.contentid , f.detaildecide,f.detailscore from bscheckmaster e ,bscheckdetail f ");

sb.append("where e.resultid = f.resultid and e.resultstate<>4 and e.resultid=:resultid and e.planid= :planid and e.comid= :comid and 1=1 ");

}

}

sb.append(" ) h on g.contentid = h.contentid order by g.itemsortid,g.contentsortid");

String sql = sb.toString();

// 转化sql语句

String[] paramName = **new** String[] { "planid", "itemid","resultid", "planid",

"comid" };

//检查总数

/\*\*

\*计划检查项总数

\* **@param** request

\* **@param** dto

\* **@return**

\* **@throws** Exception

\*/

**public** **int** getplanChecknum(BscheckplanDTO dto ) **throws** Exception {

//使用字符串缓冲器类拼接查询语句

Cnd wh = **null**;

//转化sql语句

**int** li\_count= dao.count(Bscheckpicset.**class**, wh.*where*("planid", "=", dto.getPlanid().trim()));

**return** li\_count;

}

//实际检查总数

/\*\*

结果检查项总数 \*/

**public** **int** getresultChecknum(BscheckplanDTO dto ) **throws** Exception {

//使用字符串缓冲器类拼接查询语句

StringBuffer sb = **new** StringBuffer();

sb.append(" select \* from bscheckdetail c , (select l.resultid from bscheckmaster l , (select MAX(a.resultdate) as resultdate ");

//sb.append(" a.resultid from bscheckmaster a where a.planid= :planid and a.comid= :comid");

sb.append(" from bscheckmaster a where a.planid= :planid and a.comid= :comid");

sb.append(" ) p where l.resultdate = p.resultdate ) j where c.resultid= j.resultid ");

String sql = sb.toString();

//转化sql语句

String[] paramName = **new** String[]{"planid","comid"};

**int**[] paramType = **new** **int**[]{Types.*VARCHAR*,Types.*VARCHAR*};

sql = QueryFactory.*getSQL*(sql, paramName, paramType, dto);

Map m = DbUtils.*DataQuery*(GlobalNames.*sql*, sql, **null**, Map.**class**);

**int** total = (Integer) m.get(GlobalNames.*SI\_TOTALROWNUM*);

**return** total;

}

1. 如果检查完成

Intent intent = getIntent();

resultid = intent.getStringExtra("resultid");

path = Contances.*showResultPrintUrl* + "?resultid=" + resultid;

// 核查意见及结果的接口

**public** **static** String *showResultPrintUrl* = *baseUrl* + "getResultHtmlByresultId";

/\*\*

\*

\* getResultHtmlByresultId的中文名称：根据结果id获取结果明细页面信息

\*

\* getResultHtmlByresultId的概要说明：传入resultid

\*

\* **@param** dto

\* **@return** Written by : tmm

\*

\*/

@At

@Ok("raw:htm")

**public** Object getResultHtmlByresultId(HttpServletRequest request, @Param("..") BsCheckMasterDto dto) {

**try** {

**return** sjbService.getResultDetail(request, dto).getDetailinfo();

} **catch** (Exception e) {

**return** "暂无数据";

}}

/\*\*

\* 查询出结果对象

\*

\* **@param**

\* **@return**

\* **@throws** Exception

\*/

**public** BsCheckMaster getResultDetail(HttpServletRequest request,

BsCheckMasterDto dto) **throws** Exception {

BsCheckMaster master = dao

.fetch(BsCheckMaster.**class**, dto.getResultid());

StringBuffer sb = **new** StringBuffer();

sb.append("<!DOCTYPE html><html> <head><title>结果明细</title><meta http-equiv='content-type' content='text/html; charset=UTF-8'>");

sb.append("<meta name='Author' content=''> <meta name='Keywords' content=''> <meta name='Description' content=''>");

sb.append("</head><body>");

**if** (master != **null**) {

Bscheckplan plan = dao

.fetch(Bscheckplan.**class**, master.getPlanid());

// gu20161114 Pcompany pcompany = dao

// .fetch(Pcompany.class, master.getComid());

**if**(plan!=**null**){

String type = plan.getPlanchecktype();

**if**("0".equals(type)){

//量化明细表

// sb.append(master.getCheckresultinfo());

sb.append(master.getDetailinfo());

}**else** **if**("1".equals(master.getCheckgroupstate())) {

//日常结果记录表

sb.append(master.getDetailinfo());

}**else** {

//日常结果记录表

sb.append(master.getCheckresultinfo());

sb.append(master.getDetailinfo());

// if(pcompany!=null){//企业（食品生产显示明细）

// if("1".equals(pcompany.getComdalei())){//食品生产

// sb.append(master.getDetailinfo());

// }

// }

}

}

} **else** {

master = **new** BsCheckMaster();

sb.append("暂无数据");

}

sb.append("</body></html>");

master.setDetailinfo(sb.toString());

**return** master;

}

保存按钮代码

**private** ArrayList<DailySupervision> dailySupervision;

从下面方法获取的

**private** **void** initData() {

Intent intent = getIntent();

planid = intent.getStringExtra("planid");

itemid = intent.getStringExtra("itemid");

resultid = intent.getStringExtra("resultid");

String resultstate = intent.getStringExtra("resultstate") + "";

**if** (resultstate.equals("1") || resultstate.equals("2") || resultstate.equals("3")) {

bt\_all\_select\_match.setVisibility(View.*GONE*);

} **else** {

bt\_all\_select\_match.setVisibility(View.*VISIBLE*);

}

System.*out*.println(resultid + "////////////////////");

comid = sp.getString("comid", "");

aaa383 = sp.getString("aaa383", "");

aaa027 = sp.getString("aaa027", "");

String userName = sp.getString("userName", "");

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

tv\_check\_applicant.setText(userName);

}

// final String itemid = sp.getString("itemid", "");

System.*out*.println(planid + "=======" + itemid + "======" + comid);

// **TODO** 联网操作

**new** Thread() {

**public** **void** run() {

**try** {

// **TODO** 获取结果ID接口

**final** String path = Contances.*DailySupervisionUrl* + "?itemid=" + itemid + "&planid=" + planid

+ "&comid=" + comid + "&resultid=" + resultid;

System.*out*.println(comid);

URL url = **new** URL(path);

HttpURLConnection conn = (HttpURLConnection) url.openConnection();

conn.setRequestMethod("GET");

conn.setConnectTimeout(30000);

**int** code = conn.getResponseCode();

**if** (code == 200) {

// 联网成功

InputStream is = conn.getInputStream();

**final** String info = StreamTools.*getStringFromStream*(is);

Gson gson = **new** Gson();

DailySupervisionBean dailySupervisionBean = gson.fromJson(info, DailySupervisionBean.**class**);

dailySupervision = dailySupervisionBean.rows;

runOnUiThread(**new** Runnable() {

**public** **void** run() {

**if** (dailySupervision != **null**) {

adapter = **new** MyAdapter(ctx);

lv\_scene\_check.setAdapter(adapter);

lv\_scene\_check.setVisibility(View.*VISIBLE*);

}

}

});

} **else** {

showToast("服务器忙,请稍后...");

}

} **catch** (SocketException e) {

showToast("数据加载失败,请检查网络...");

e.printStackTrace();

} **catch** (Exception e) {

e.printStackTrace();

}

}

}.start();

}

bt\_save\_data.setOnClickListener(**new** OnClickListener() {

@Override

**public** **void** onClick(View v) {

**try** {

**int** j = 0;

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

**if** (dailySupervision.get(i).getFlag() != 0) {

// 明细得分

System.*out*.println(dailySupervision.get(i).getFlag() + "--------------" + i);

String contentscore = dailySupervision.get(i).contentscore;

clientKey = **new** JSONObject();

clientKey.put("contentid", dailySupervision.get(i).contentid);

clientKey.put("detailscore", contentscore);

clientKey.put("detaildecide", dailySupervision.get(i).getFlag());

list.put(j, clientKey);

j++;

System.*out*.println("------------------" + j);

}

/\* 封装数组 \*/

}

} **catch** (JSONException e) {

// **TODO** Auto-generated catch block

e.printStackTrace();

}

**try** {

**if** (list.length() != 0) {

params.put("resultid", resultid);

params.put("comdalei", comdalei);

// 保存

params.put("resultstate", 2);

params.put("aaa027", aaa027);

params.put("aaa383", aaa383);

params.put("detaillist", list);

// params.put("userid", userid);

**if** (**null** == pDialog) {

pDialog = **new** ProgressDialog(ctx);

}

pDialog.setMessage("正在保存数据，请稍后...");// 设置对话框中的提示信息

pDialog.show();

} **else** {

showToast("请选择检查项");

**return**;

}

System.*out*.println(list.toString() + "--------------");

} **catch** (JSONException e) {

e.printStackTrace();

}

String toast = "保存成功，请提交数据...";

**int** saveOrSubmit = 1001;

saveResult(Contances.*SaveDetailUrl*, toast, saveOrSubmit);

**try** {

Thread.*sleep*(500);

} **catch** (InterruptedException e) {

// **TODO** Auto-generated catch block

e.printStackTrace();

}

}

});

// 查询检查保存数据---现场核查

**public** **static** String *SaveDetailUrl* = *baseUrl* + "bathSaveCheckDetail";

/\*\*

\*

\* bathSaveCheckDetail的中文名称：保存检查结果明细

\*

\* bathSaveCheckDetail的概要说明：

\*

\* **@param** request

\* **@return**

\* **@throws** Exception

\* Written by : tmm

\*/

@AdaptBy(type = JsonAdaptor.**class**)

@At

@Ok("json")

**public** Object bathSaveCheckDetail(HttpServletRequest request, @Param("..") BsCheckDetailDto dto, @Param("..") BsCheckMasterDto mdto)

**throws** Exception {

// request.getParameter(arg0);

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

result.put("code", "9999");

result.put("msg", "保存失败");

result.put("success", **false**);

// 批量保存明细

Map map = checkHandlerService.bathsaveCheckDetail(request, dto);

// 修改结果状态(保存平均分)

mdto.setResultid(dto.getResultid());

sjbService.updateResultStateByresultid(request, mdto);

**if** ((Boolean) map.get("success")) {

result.put("msg", map.get("msg"));

result.put("success", map.get("success"));

result.put("code", "0000");

} **else** {

result.put("msg", map.get("msg"));

}

**return** result;

}

/\*\*

\* saveCheckDetail的中文名称：保存检查结果明细

\* saveCheckDetail的概要说明：

\* **@param** dto

\* **@return** Written by : syf

\* **@throws** Exception

\*

\*/

@Aop( { "trans" })

**public** Map bathsaveCheckDetail(HttpServletRequest request, BsCheckDetailDto dto) **throws** Exception {

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

**try**{

**for**(BsCheckDetailDto detail : dto.getDetaillist())

{

BsCheckDetail detailbean = **new** BsCheckDetail();

detailbean.setContentid(detail.getContentid());

detailbean.setResultid(dto.getResultid());

detailbean.setDetaildecide(detail.getDetaildecide());

//符合，填写分数

**if**("1".equals(detail.getDetaildecide())){

// System.out.println(detail.getDetailscore());

detailbean.setDetailscore(detail.getDetailscore());

}

saveCheckDetail(request, detailbean);

}

retMap.put("success", **true**);

retMap.put("msg", "保存成功");

}**catch** (Exception e) {

retMap.put("success", **false**);

retMap.put("msg", "保存失败");

}

**return** retMap;

}

/\*\*

\*

\* saveCheckDetail的中文名称：保存检查结果明细

\* saveCheckDetail的概要说明：

\*

\* **@param** dto

\* **@return** Written by : syf

\* **@throws** Exception

\*

\*/

@Aop( { "trans" })

**public** Map saveCheckDetail(HttpServletRequest request, BsCheckDetail dto) **throws** Exception {

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

retMap.put("success", **false**);

retMap.put("msg", "操作失败");

retMap.put("detailid", "");

**if**(StringUtils.*isEmpty*(dto.getResultid())){

retMap.put("msg", "检查结果ID[resultid]不能为空");

**return** retMap;

}

**if**(StringUtils.*isEmpty*(dto.getContentid())){

retMap.put("msg", "检查内容ID[contentid]不能为空");

**return** retMap;

}

// Sysuser user = SysmanageUtil.getSysuser();

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

dto.setDetailoperatedate(**new** Date());

dto.setDetailcheckdate(**new** Date());

// dto.setDetailcheckperson(user.getUserid().toString());

dto.setDetailoperatedate(**new** Date());

// dto.setDetailoperateperson(user.getUserid().toString());

BsCheckDetail detail = queryResultDetailISExist(request,dto);

**if** (**null** != detail) {

detail.setDetaildecide(dto.getDetaildecide());

//符合之外，分数为0

**if**(!"1".equals(dto.getDetaildecide())){

detail.setDetailscore(0);

}**else**{

detail.setDetailscore(dto.getDetailscore());

}

detail.setDetailng(dto.getDetailng());

detail.setDetailattachment(dto.getDetailattachment());

detail.setDetailremark(dto.getDetailremark());

**if**(dao.update(detail)>0){

retMap.put("success", **true**);

retMap.put("msg", "修改成功");

retMap.put("detailid", dto.getDetailid());

}

} **else** {

String sequence = DbUtils.*getSequenceStr*();

dto.setDetailid(sequence);

//符合之外，分数为0

**if**(!"1".equals(dto.getDetaildecide())){

dto.setDetailscore(0);

}

dto = dao.insert(dto);

retMap.put("success", **true**);

retMap.put("msg", "保存成功");

retMap.put("detailid", dto.getDetailid());

}

**return** retMap;

}

/\*\*

\* 查询检查明细结果 结果id，(手机调用)

\*

\* **@param** resultid

\* 计划id

\* **@return**

\* **@throws** Exception

\*/

**public** Map updateResultStateByresultid(HttpServletRequest request,

BsCheckMasterDto dto) **throws** Exception {

// 使用字符串缓冲器类拼接查询语句

Map map = **new** HashMap();

StringBuffer sb = **new** StringBuffer();

**try** {

sb.append(" select b.planchecktype,a.\* ");

sb.append("from bscheckmaster a ,bscheckplan b ");

sb.append(" where a.planid=b.planid ");

sb.append("and a.resultid =:resultid ");

sb.append(" order by a.planid ");

String sql = sb.toString();

// 转化sql语句

String[] paramName = **new** String[] { "resultid"};

**int**[] paramType = **new** **int**[] { Types.*VARCHAR* };

sql = QueryFactory.*getSQL*(sql, paramName, paramType, dto);

System.*out*.println(sql);

Map m = DbUtils.*DataQuery*(GlobalNames.*sql*, sql, **null**, BsCheckMasterDto.**class**);

List<BsCheckMasterDto> list = (List) m.get(GlobalNames.*SI\_RESULTSET*);

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

map.put("msg", "没有相关数据无法进行此操作");

map.put("success", **false**);

} **else** {// 修改结果状态

BsCheckMasterDto masterold = list.get(0);

BsCheckMaster master = **new** BsCheckMaster();

BeanHelper.*copyProperties*(masterold, master);

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

master.setResultstate(dto.getResultstate());// 结束检查

master.setResultdate(**new** Date());// 结束时间

master.setOperatedate(sdf.parse(masterold.getOperatedate()));//经办时间

//检查类型

dto.setPlanchecktype(masterold.getPlanchecktype());

dto.setCheckgroupstate(masterold.getCheckgroupstate());

**if**("3".equals(dto.getResultstate())){//明细固化

String result = SysmanageUtil.*getAA027Str*(dto.getAaa027(), dto.getAaa383());

dto.setAaa027(result);

master.setDetailinfo((String)getDetailByid(request, dto).get("data"));

}

**if**("0".equals(dto.getPlanchecktype())){

BsCheckMasterDto checkMaster = getAverageScore(dto);

**if**(checkMaster!=**null**){

//得分

master.setResultscore(checkMaster.getChecknum());

//平均分

// master.setResultscore(checkMaster.getChecknum());

//等级

// String level = getlevelInfo(checkMaster.getAverageScore(), dto);

}

}

dao.update(master);

map.put("msg", "保存成功");

// map.put("data", getDetailByid(request, dto).get("data"));

map.put("success", **true**);

}

} **catch** (Exception e) {

e.printStackTrace();

map.put("success", **false**);

map.put("error", Lang.*wrapThrow*(e).getMessage());

}

**return** map;

}

/\*\*

\* 查询检查明细结果 结果id，

\* **@param** resultid 计划id

\* **@return**

\* **@throws** Exception

\*/

**public** Map getDetailByid(HttpServletRequest request,BsCheckMasterDto dto) **throws** Exception {

//使用字符串缓冲器类拼接查询语句

StringBuffer sb = **new** StringBuffer();

String str ="";

sb.append(" select b.contentcode,b.contentimpt ,b.contentsortid,b.contentscore ,c.itemsortid sortid, ");

sb.append(" a.resultid , b.contentid , b.content ,c.itemname ,c.itemid,c.itempid , ");

sb.append(" a.detaildecide , a.detailscore ,a.detailng ,a.detailremark, ");

sb.append(" (select COUNT(\*) from bscheckdetail d ,omcheckcontent e ");

sb.append(" where d.contentid=e.contentid and e.itemid=c.itemid and d.resultid = :resultid ) count ");

sb.append(" from bscheckdetail a , omcheckcontent b , omcheckgroup c ");

sb.append(" where a.contentid = b.contentid and b.itemid = c.itemid ");

sb.append(" and a.resultid= :resultid ");

sb.append(" ORDER BY sortid ,b.contentsortid");

String sql = sb.toString();

//转化sql语句

String[] paramName = **new** String[]{"resultid"};

**int**[] paramType = **new** **int**[]{Types.*VARCHAR*};

sql = QueryFactory.*getSQL*(sql, paramName, paramType, dto);

System.*out*.println("sql "+sql);

Map m = DbUtils.*DataQuery*(GlobalNames.*sql*, sql, **null**, BsCheckDetailDto.**class**);

List<BsCheckDetailDto> list = (List<BsCheckDetailDto>) m.get(GlobalNames.*SI\_RESULTSET*);

// 拼接数据(通过企业大类类区分表头拼接不同的明细表)

**if**("2".equals(dto.getComdalei())){//药品生产企业（死数据）

str = getYPDetaillist( request, dto,list,"YPSC");

}**else** **if**("8".equals(dto.getComdalei())){//药品批发

str = getYPDetaillist( request, dto,list,"YPJY");

}**else** **if**("9".equals(dto.getComdalei())){//药品零售

str = getYPDetaillist( request, dto,list,"YPJY");

}**else** **if**("1".equals(dto.getCheckgroupstate())){//聚餐检查

str = getJCSBDetaillist( request, dto,list);

}**else** **if**("19".equals(dto.getComdalei()) || "3".equals(dto.getComdalei())){//保健食品

str = getDetaillist( request, dto,list,"BJP");

}

**else** {

str = getDetaillist( request, dto,list,**null**);

}

Map map = **new** HashMap();

map.put("data", str);

**return** map;

}

/\*\*

\* 食品拼接明细字符串

\* **@param** list 明细列表

\* **@return**

\* **@throws** Exception

\*/

**public** String getDetaillist(HttpServletRequest request,BsCheckMasterDto dto,List<BsCheckDetailDto> list, String state) **throws** Exception {

List itemidlist = **new** ArrayList();

StringBuffer sb = **new** StringBuffer();

String tbody="";

String tbodyBotoom="";

String zfryzjhm="";

String tbodyOne="";//☑

// String checkTrue ="<input type='checkbox' style='height:20px;width:20px;' checked='checked'>";

String checkTrue ="☑";

String checkFalse ="□";//&#9633;□

**if**("BJP".equals(state)){

dto.setTbodytype("bjp\_checkPaln");

}**else** {

dto.setTbodytype("checkPaln");

}

dto.setTbodycode(dto.getPlanchecktype());//计划类别

//获取执法人员信息

zfryzjhm= getzfzjh(dto.getUserid()); //执法证件号

//获取企业信息

PcompanyDTO company = getQiyeInfoByid(request, dto);

//获取表头信息

BsTbodyInfoDTo tbodybean = getTbodyInfo(request, dto);

**if**(tbodybean!=**null**){

tbody= tbodybean.getTbodyinfo();

tbodyBotoom=tbodybean.getTfootinfo();

tbodyOne=tbodybean.getTbody();

}

String tbodyval="";

//表头

//量化

**if**("0".equals(dto.getPlanchecktype())){

**if**(company!=**null** ){

String starttime = company.getOperatedate();

String endtime = company.getResultdate();

//表头赋值

**if**(starttime.substring(0,10) == endtime.substring(0,10)){//同一天

tbodyval=tbody.replace("commc",company.getCommc()==**null**?"":company.getCommc())

.replace("comdz",company.getComdz()==**null**?"":company.getComdz())

.replace("comfrhyz",company.getComfrhyz()==**null**?"":company.getComfrhyz())

.replace("comyddh",company.getComyddh()==**null**?"":company.getComyddh())

// .replace("comfrhyz",dto.getLxr()==null?"":dto.getLxr())

// .replace("comyddh",dto.getLxdh()==null?"":dto.getLxdh())

.replace("zfryzjhm", zfryzjhm==**null**?"":zfryzjhm)

.replace("comxkzbh",company.getComxkzbh()==**null**?"":company.getComxkzbh())

.replace("comdmlx",company.getComdmlx()==**null**?"":company.getComdmlx())//店面

.replace("check\_year",starttime.substring(0,4)==**null**?"":starttime.substring(0,4))

.replace("check\_month",starttime.substring(5,7)==**null**?"":starttime.substring(5,7))

.replace("check\_data",starttime.substring(8,10)==**null**?"":starttime.substring(8,10))

.replaceFirst("check\_time",starttime.substring(11,13)==**null**?"":starttime.substring(11,13))

.replaceFirst("check\_fen",starttime.substring(14,16)==**null**?"":starttime.substring(14,16))

.replace("to\_check\_time",endtime.substring(11,13)==**null**?"":endtime.substring(11,13))

.replace("to\_check\_fen",endtime.substring(14,16)==**null**?"":endtime.substring(14,16));

}**else**{

tbodyval= tbody.replace("commc",company.getCommc()==**null**?"":company.getCommc())

.replace("comdz",company.getComdz()==**null**?"":company.getComdz())

.replace("comfrhyz",company.getComfrhyz()==**null**?"":company.getComfrhyz())

.replace("comyddh",company.getComyddh()==**null**?"":company.getComyddh())

// .replace("comfrhyz",dto.getLxr()==null?"":dto.getLxr())

// .replace("comyddh",dto.getLxdh()==null?"":dto.getLxdh())

.replace("zfryzjhm", zfryzjhm==**null**?"":zfryzjhm)

.replace("comxkzbh",company.getComxkzbh()==**null**?"":company.getComxkzbh())

.replace("comdmlx",company.getComdmlx()==**null**?"":company.getComdmlx())

.replace("check\_year",starttime.substring(0,4)==**null**?"":starttime.substring(0,4))

.replace("check\_month",starttime.substring(5,7)==**null**?"":starttime.substring(5,7))

.replace("check\_data",starttime.substring(8,10)==**null**?"":starttime.substring(8,10))

.replaceFirst("check\_time",starttime.substring(11,13)==**null**?"":starttime.substring(11,13))

.replaceFirst("check\_fen",starttime.substring(14,16)==**null**?"":starttime.substring(14,16))

.replace("to\_check\_time",endtime.substring(0,13)==**null**?"":endtime.substring(0,13))

.replace("to\_check\_fen",endtime.substring(14,16)==**null**?"":endtime.substring(14,16));

}

}

//固定表头

**if**(tbodyOne!=**null**&&!"".equals(tbodyOne)){

String str = tbodyOne.replace("</tbody>", "").replace("</table>", "");

sb.append(str);

}

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

//含有不合并

**if**(itemidlist.contains(list.get(i).getItemid())){

sb.append("<tr><td>"+list.get(i).getContentcode()+"."+list.get(i).getContent()+"</td><td style='text-align:center;'>"+list.get(i).getContentscore()+"</td><td style='text-align:center;'>"+list.get(i).getDetailscore()+"</td></tr>");

}**else**{

itemidlist.add(list.get(i).getItemid());

sb.append("<tr ><td id="+list.get(i).getItemid()+" rowspan="+list.get(i).getCount()+" style='text-align:center;' >"+list.get(i).getItemname()+"</td><td>"+list.get(i).getContentcode()+"."+list.get(i).getContent()+"</td><td style='text-align:center;'>"+list.get(i).getContentscore()+"</td><td style='text-align:center;'>"+list.get(i).getDetailscore()+"</td></tr>");

}

}

//平均分

**float** averageScore =0;

BsCheckMasterDto checkMaster = getAverageScore(dto);

**if**(checkMaster!=**null**){

averageScore=checkMaster.getAverageScore();

}

//等级

String level= getlevelInfo(averageScore, dto);

//格式化小数，不足的补0

DecimalFormat df = **new** DecimalFormat("0.0");

String average = df.format(averageScore);

String[] str =level.split("&");

tbodyBotoom=tbodyBotoom.replace("average", average).replace("assesslevel", str[0]);

sb.append(tbodyBotoom);

tbodyBotoom="";

sb.append("</tbody></table>");

}**else** **if**("1".equals(dto.getPlanchecktype())){//日常

//表头赋值

**if**(company!=**null** ){

tbodyval=tbody.replace("Itemname",company.getItemname()==**null**?"":company.getItemname())

.replace("commc",company.getCommc()==**null**?"":company.getCommc())

.replace("comdz",company.getComdz()==**null**?"":company.getComdz())

.replace("zfryzjhm", zfryzjhm==**null**?"":zfryzjhm)

.replaceFirst("check\_startime",company.getOperatedate()==**null**?"":DateUtil.*getDateString*("yyyy-MM-dd HH:mm:ss","yyyy年MM月dd日 HH时mm分ss秒",company.getOperatedate()))

.replace("check\_endtime",company.getResultdate()==**null**?"":DateUtil.*getDateString*("yyyy-MM-dd HH:mm:ss","yyyy年MM月dd日 HH时mm分ss秒",company.getResultdate()))

.replace("checkdz",company.getComdz()==**null**?"":company.getComdz());

// .replace("</tbody>", "").replace("</table>", "");

}

**if**(tbodyOne!=**null**&&!"".equals(tbody)){

String str = tbodyOne.replace("Itemname",company.getItemname()==**null**?"":company.getItemname())

.replace("</tbody>", "").replace("</table>", "");

sb.append(str);

}

//明细拼接

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

//含有不合并

**if**(itemidlist.contains(list.get(i).getItemid())){

// sb.append("<tr id='tr"+i+"' ><td>"+list.get(i).getContentsortid()+"</td><td>"+list.get(i).getContent()+"</td><td>"+getconentimpinfo("CONTENTIMPT",list.get(i).getContentimpt())+"</td><td>"+(("1".equals(list.get(i).getDetaildecide()))?"&radic;":"")+"</td><td>"+(("2".equals(list.get(i).getDetaildecide()))?"&radic;":"")+"</td><td>"+(("3".equals(list.get(i).getDetaildecide()))?"&radic;":"")+"</td></tr>");

sb.append("<tr id='tr"+i+"' ><td style='text-align:center;'>"+list.get(i).getContentcode()+"</td><td>"+list.get(i).getContent()+"</td><td style='text-align:center;'>"+(("1".equals(list.get(i).getDetaildecide()))?checkTrue:checkFalse)+"是&nbsp;"+(("2".equals(list.get(i).getDetaildecide()))?checkTrue:checkFalse)+"否</td><td style='text-align:center;'>"+(("3".equals(list.get(i).getDetaildecide()))?"不适用":"")+"</td></tr>");

}**else**{

itemidlist.add(list.get(i).getItemid());

// sb.append("<tr id='total"+i+"' ><td id="+list.get(i).getItemid()+" rowspan="+list.get(i).getCount()+">"+list.get(i).getItemname()+"</td><td>"+list.get(i).getContentsortid()+"</td><td>"+list.get(i).getContent()+"</td><td>"+getconentimpinfo("CONTENTIMPT",list.get(i).getContentimpt())+"</td><td>"+(("1".equals(list.get(i).getDetaildecide()))?"&radic;":"")+"</td><td>"+(("2".equals(list.get(i).getDetaildecide()))?"&radic;":"")+"</td><td>"+(("3".equals(list.get(i).getDetaildecide()))?"&radic;":"")+"</td></tr>");

sb.append("<tr id='total"+i+"' ><td id="+list.get(i).getItemid()+" rowspan="+list.get(i).getCount()+">"+list.get(i).getItemname()+"</td><td style='text-align:center;' >"+list.get(i).getContentcode()+"</td><td>"+list.get(i).getContent()+"</td><td style='text-align:center;'>"+(("1".equals(list.get(i).getDetaildecide()))?checkTrue:checkFalse)+"是&nbsp;"+(("2".equals(list.get(i).getDetaildecide()))?checkTrue:checkFalse)+"否</td><td style='text-align:center;'>"+(("3".equals(list.get(i).getDetaildecide()))?"不适用":"")+"</td></tr>");

}

}

sb.append("</tbody></table>");

}

**return** tbodyval+sb.toString()+tbodyBotoom;

}

/\*\*

\* 查询企业信息通过结果id，

\* **@param** 结果id

\* **@return**

\* **@throws** Exception

\*/

**public** PcompanyDTO getQiyeInfoByid(HttpServletRequest request,BsCheckMasterDto dto) **throws** Exception {

//使用字符串缓冲器类拼接查询语句

StringBuffer sb = **new** StringBuffer();

sb.append(" select DISTINCT (select e.itemname from omcheckgroup e where e.itemid=d.itempid) itemname, b.\*,a.operatedate,a.resultdate");

sb.append(" from bscheckmaster a ,pcompany b ,bscheckpicset c ,omcheckgroup d ");

sb.append(" where a.comid=b.comid and c.planid= a.planid and c.itemid = d.itemid ");

sb.append(" and a.resultid= :resultid ");

String sql = sb.toString();

//转化sql语句

String[] paramName = **new** String[]{"resultid"};

**int**[] paramType = **new** **int**[]{Types.*VARCHAR*};

sql = QueryFactory.*getSQL*(sql, paramName, paramType, dto);

System.*out*.println(sql);

Map m = DbUtils.*DataQuery*(GlobalNames.*sql*, sql, **null**, PcompanyDTO.**class**);

List<PcompanyDTO> list = (List<PcompanyDTO>) m.get(GlobalNames.*SI\_RESULTSET*);

**if**(list.size()>0){

**return** list.get(0);

}

**return** **null**;

}

/\*\*

\* 得到表头信息，

\* **@param** 结果id

\* **@return**

\* **@throws** Exception

\*/

**public** BsTbodyInfoDTo getTbodyInfo(HttpServletRequest request,BsCheckMasterDto dto) **throws** Exception {

//使用字符串缓冲器类拼接查询语句

StringBuffer sb = **new** StringBuffer();

sb.append(" select a.tbodyinfo ,a.tfootinfo,a.tbody from bstbodyinfo a ");

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

sb.append(" and a.tbodytype = :tbodytype and a.tbodycode= :tbodycode and a.aaa027 like :aaa027 ");

String sql = sb.toString();

//转化sql语句

String[] paramName = **new** String[]{"tbodytype","tbodycode","aaa027"};

**int**[] paramType = **new** **int**[]{Types.*VARCHAR*,Types.*VARCHAR*,Types.*VARCHAR*};

sql = QueryFactory.*getSQL*(sql, paramName, paramType, dto);

System.*out*.println(sql);

Map m = DbUtils.*DataQuery*(GlobalNames.*sql*, sql, **null**, BsTbodyInfoDTo.**class**);

List<BsTbodyInfoDTo> list = (List<BsTbodyInfoDTo>) m.get(GlobalNames.*SI\_RESULTSET*);

**if**(list.size()>0){

**return** list.get(0);

}

**return** **null**;

}

/\*\*

\*得到量化平均分

\* **@param** request

\* **@param** dto

\* **@return**

\* **@throws** Exception

\*/

**public** BsCheckMasterDto getAverageScore(BsCheckMasterDto dto ) **throws** Exception {

//使用字符串缓冲器类拼接查询语句

StringBuffer sb = **new** StringBuffer();

**float** averageScore=0;

BsCheckMasterDto checkMaster = **null**;

sb.append(" select sum(a.detailscore) as checknum,sum(b.contentscore) as checkgetnum from ");

sb.append(" bscheckdetail a , omcheckcontent b where a.contentid = b.contentid");

sb.append(" and a.resultid= :resultid and 1=1 ");

sb.append(" ORDER BY a.detailscore ");

String sql = sb.toString();

//转化sql语句

String[] paramName = **new** String[]{"resultid"};

**int**[] paramType = **new** **int**[]{Types.*VARCHAR*};

sql = QueryFactory.*getSQL*(sql, paramName, paramType, dto);

Map m = DbUtils.*DataQuery*(GlobalNames.*sql*, sql, **null**, BsCheckMasterDto.**class**);

List<BsCheckMasterDto> list = (List<BsCheckMasterDto>) m.get(GlobalNames.*SI\_RESULTSET*);

**if**(list.size()>0){

checkMaster = list.get(0);

// //实际得到分数

**float** checkNum = checkMaster.getChecknum();

// //总分

**float** checkgetNum = checkMaster.getCheckgetnum();

averageScore = checkNum/checkgetNum\*10;

checkMaster.setAverageScore(averageScore);

// DecimalFormat df = new DecimalFormat("0.0");//格式化小数，不足的补0

// averageScore = df.format(score);

}

**return** checkMaster;

}

/\*\*

\* 量化评定等级公式(评定分数在9.0分以上（含9.0分），为优秀；评定分数在8.9分至7.5分（含7.5分），为良好；评定分数在7.4分至6.0分（含6.0分），为一般。)

\* 评定分数在6.0分以下的，或2项以上（含2项）关键项不符合要求的，且情形不属于食物中毒或应受到行政处罚的，

\* 不评定动态等级。在等级公示牌“本次检查动态等级”处张贴“整改中”字样。

\* **@param** request

\* **@param** dto

\* **@return**

\* **@throws** Exception

\*/

**public** String getlevelInfo(**double** value,BsCheckMasterDto dto ) **throws** Exception {

String level="";

String levelGrapheme="";

dto.setNum(1);

dto.setDetaildecide("2");

//关键项不合格个数

**int** impcount=getQualifiedNum(**null**, dto) ;

**if**(impcount>=2){

level="整改中";

levelGrapheme="整改中";

}**else**{

**if**(value>=9.0){

level="优秀";

levelGrapheme="A";

}**else** **if**(value>=7.5 && value<8.9){

level="良好";

levelGrapheme="B";

}**else** **if**(value>=6.0 && value<7.4){

level="一般";

levelGrapheme="C";

}**else** {

level="整改中";

levelGrapheme="整改中";

}

}

**return** level+"&"+levelGrapheme;

}

/\*\*

\* 通过查询合格明细个数(传入detaildecide和重要性num )

\*

\* **@param**

\* **@return**

\* **@throws** Exception

\*/

**public** **int** getQualifiedNum(HttpServletRequest request, BsCheckMasterDto dto)

**throws** Exception {

// 使用字符串缓冲器类拼接查询语句

StringBuffer sb = **new** StringBuffer();

sb.append(" bscheckdetail a, omcheckcontent b ");

sb.append(" where a.contentid=b.contentid and a.resultid=:resultid");

sb.append(" and a.detaildecide =:detaildecide and b.contentimpt <> :num");

sb.append(" and 1=1 ");

sb.append(" order by a.resultid ");

String sql = sb.toString();

// 转化sql语句

String[] paramName = **new** String[] { "detaildecide" ,"num","resultid"};

**int**[] paramType = **new** **int**[] { Types.*VARCHAR*,Types.*INTEGER*,Types.*VARCHAR* };

sql = QueryFactory.*getSQL*(sql, paramName, paramType, dto);

**int** total = dao.count(sql);

**return** total;

}

提交按钮

// 提交

bt\_submit\_data.setOnClickListener(**new** OnClickListener() {

@Override

**public** **void** onClick(View v) {

**try** {

params.put("resultid", resultid);

params.put("comdalei", comdalei);

// 提交

params.put("aaa027", aaa027);

params.put("aaa383", aaa383); //统筹区级别

params.put("resultstate", 3);

params.put("userid", userid);

// params.put("detaillist", list);

System.*out*.println(params.toString() + "--------------");

} **catch** (JSONException e) {

// **TODO** Auto-generated catch block

e.printStackTrace();

}

**int** saveOrSubmit = 1000;

String toast = "数据提交成功";

saveResult(Contances.*SubmitDetailUrl*, toast, saveOrSubmit);

**try** {

Thread.*sleep*(500);

} **catch** (InterruptedException e) {

// **TODO** Auto-generated catch block

e.printStackTrace();

}

}

});

// 查询检查提交数据---现场核查

**public** **static** String *SubmitDetailUrl* = *baseUrl* + "updateResultByid";

/\*\*

\*

\* updateResultByid的中文名称：修改结果状态和明细信息(手机明细提交)(食品生产企业特殊判断)

\*

\* updateResultByid的概要说明：resulid(查询顺序需要调整)

\*

\* **@param** dto

\* **@return** Written by : tmm

\*

\*/

@AdaptBy(type = JsonAdaptor.**class**)

@At

@Ok("json")

**public** Object updateResultByid(HttpServletRequest request, @Param("..") BsCheckMasterDto dto) {

Map map = **new** HashMap();

**try** {

// 修改结果状态和明细信息

// 判断大类选择后台页面拼接

map = sjbService.updateResultStateByresultid(request, dto);

**return** SysmanageUtil.*execAjaxResult*(map, **null**);

} **catch** (Exception e) {

**return** SysmanageUtil.*execAjaxResult*(map, e);

}

}

**public** **void** run() {

// showUpdateDialog();

**if** (saveOrSubmit == 1001) { //保存

bt\_submit\_data.setVisibility(View.*VISIBLE*);

bt\_save\_data.setVisibility(View.*GONE*);

pDialog.dismiss();

showToast(toast);

} **else** **if** (saveOrSubmit == 1000) { //提交

bt\_submit\_data.setVisibility(View.*VISIBLE*);

bt\_save\_data.setVisibility(View.*GONE*);

pDialog.dismiss();

showToast(toast);

Intent intent = **new** Intent(ctx, CheckResultInformationActivity.**class**);

intent.putExtra("resultid", resultid);

startActivity(intent);

finish();

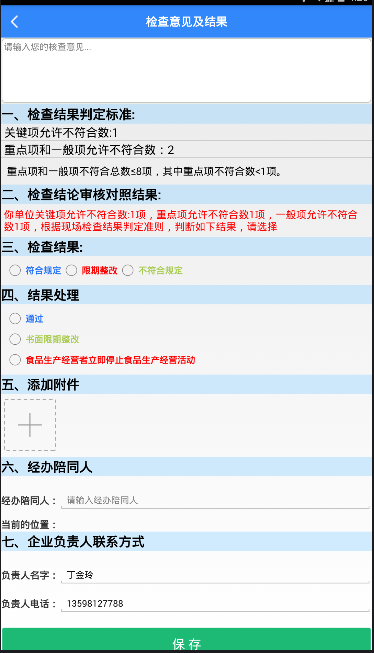
}

}

打印前填充信息界面 安全监管---检查意见及结果

**package** com.askj.tangyin.drug.app;

**public** **class** CheckResultInformationActivity **extends** Activity **implements** OnClickListener {



**package** com.askj.tangyin.drug.app;

**public** **class** CheckResultInformationActivity **extends** Activity **implements** OnClickListener {

**private** **void** initData() {

comid = sp.getString("comid", "");

aaa383 = sp.getString("aaa383", "");

aaa027 = sp.getString("aaa027", "");

// System.out.println(comid +

// "------------///////////////////////comid");

**final** String path = Contances.*checkResultDetailInfoUrl* + "?resultid=" + resultid;

**new** Thread() {

@Override

**public** **void** run() {

**try** {

URL url = **new** URL(path);

HttpURLConnection conn = (HttpURLConnection) url.openConnection();

conn.setRequestMethod("GET");

conn.setConnectTimeout(30000);

**int** code = conn.getResponseCode();

**if** (code == 200) {

InputStream is = conn.getInputStream();

String info = StreamTools.*getStringFromStream*(is);

System.*out*.println(info + "////////////////////////////////////");

Gson gson = **new** Gson();

CheckResultInformationBean checkResultInformationBean = (CheckResultInformationBean) gson

.fromJson(info, CheckResultInformationBean.**class**);

CheckResultInformationBean.CheckResultInformationInfo checkResultInformationInfo = checkResultInformationBean.data;

**final** String label1 = checkResultInformationInfo.label1;

**final** String label1\_01 = checkResultInformationInfo.label1\_01;

**final** String label1\_02 = checkResultInformationInfo.label1\_02;

**final** String label1\_03 = checkResultInformationInfo.label1\_03;

**final** String label2 = checkResultInformationInfo.label2;

**final** String label2\_01 = checkResultInformationInfo.label2\_01;

**final** String label3 = checkResultInformationInfo.label3;

**final** String label3\_01 = checkResultInformationInfo.label3\_01;

**final** String label3\_02 = checkResultInformationInfo.label3\_02;

**final** String label3\_03 = checkResultInformationInfo.label3\_03;

runOnUiThread(**new** Runnable() {

**public** **void** run() {

tv\_check\_result\_decision\_criteria.setText(label1);

tv\_key\_item\_allowed\_no\_match\_numbers.setText(label1\_01);

tv\_key\_items\_and\_common\_item\_allow\_no\_match\_numbers.setText(label1\_02);

tv\_key\_items\_and\_common\_item\_allow\_no\_match\_numbers\_content.setText(label1\_03);

tv\_verification\_conclusion\_audit\_control\_result.setText(label2);

tv\_verification\_conclusion\_audit\_control\_result\_content.setText(label2\_01);

tv\_check\_result.setText(label3);

bt\_match\_provisions.setText(label3\_01);

bt\_unmatch\_provisions.setText(label3\_02);

bt\_limit\_date\_rectification.setText(label3\_03);

}

});

}

} **catch** (Exception e) {

e.printStackTrace();

}

}

}.start();

}

// 核查意见及结果的接口

**public** **static** String *checkResultDetailInfoUrl* = *baseUrl* + "getResultinfo";

/\*\*

\*

\* resultMaster的中文名称：得到核查结果

\*

\* resultMaster的概要说明：

\*

\* **@param** dto

\* **@return** Written by : tmm

\*

\*

\*/

@At

@Ok("json")

**public** Object getResultinfo(HttpServletRequest request, @Param("..") BsCheckMasterDto dto) **throws** Exception {

Map map = **new** HashMap();

**try** {

map = sjbService.getinconformityNumid(request, dto);

**return** SysmanageUtil.*execAjaxResult*(map, **null**);

} **catch** (Exception e) {

**return** SysmanageUtil.*execAjaxResult*(map, e);

}

}

/\*\*

\* 通过查询不符合结果明细个数

\*

\* **@param**

\* **@return**

\* **@throws** Exception

\*/

**public** Map getinconformityNumid(HttpServletRequest request,

BsCheckMasterDto dto) **throws** Exception {

// 使用字符串缓冲器类拼接查询语句

StringBuffer sb = **new** StringBuffer();

Map map = **new** HashMap();

Map mapstr = **new** HashMap();

//检查的项目数

**int** resultnum = getDetailAllNum(request, dto);

//合理缺项数

dto.setDetaildecide("3");

**int** missItems = getDetailAllNum(request, dto);

//合格项目数

dto.setDetaildecide("1");

**int** goodItems = getDetailAllNum(request, dto);

// dto.setNum(3);

// 一般

// int total = getGeneralNum(request, dto);

// 重要和关键

// int totalimpt = getImportanceNum(request, dto);

// int num = total + totalimpt;

//合格率

// double good = goodItems/(resultnum-missItems)\*100;

//重要项不合格数

dto.setDetaildecide("2");

dto.setNum(1);

**int** impfailitems = getQualifiedNum(request, dto);

//一般项不合格数

**int** generalItems = getDetailAllNum(request, dto);

//结果状态返给手机做判断

String state ="";

//判断结果是否符合：

// state(1代表符合；2.代表不符合)

//没有不合格的或是发现检查的重点项存在1项及以下不合格且70%≤一般项合格率＜100%

// if((impfailitems==0 && generalItems==0) || (impfailitems<=1 && good<100 && good>=70)){

// state="1";

// }else if(impfailitems>=2 || good<70){

// state="2" ;

// }

//符合不符合（生产经营）

**if**(generalItems<=8){//一般项《=8 符合

state="1";

}**else** **if**(generalItems>8 ||impfailitems>=1 ){//不符合

state="2";

}

// 拼接

// 拼接

sb.append("你单位");

sb.append("" + BsCheckResultStaticDto.*LABEL\_gjxbfhs* + impfailitems + "")

.append("项，");

sb.append("" + BsCheckResultStaticDto.*LABEL\_zdxbfhs* + impfailitems + "")

.append("项，");

sb.append("" + BsCheckResultStaticDto.*LABEL\_ybxbfhs* + generalItems + "").append(

"项，");

sb.append("" + BsCheckResultStaticDto.*LABEL2\_01* + "");

// 信息传递

mapstr.put("label1", BsCheckResultStaticDto.*LABEL*);

mapstr.put("label1\_01", BsCheckResultStaticDto.*LABEL\_01* + impfailitems);

mapstr.put("label1\_02", BsCheckResultStaticDto.*LABEL\_03* + (impfailitems+generalItems));

mapstr.put("label1\_03", BsCheckResultStaticDto.*LABEL\_05*);

mapstr.put("label2", BsCheckResultStaticDto.*LABEL2*);

mapstr.put("label2\_01", sb.toString());

mapstr.put("label3", BsCheckResultStaticDto.*LABEL3*);

mapstr.put("label3\_01", BsCheckResultStaticDto.*LABEL3\_01*);

mapstr.put("label3\_02", BsCheckResultStaticDto.*LABEL3\_02*);

mapstr.put("label3\_03", BsCheckResultStaticDto.*LABEL3\_03*);

mapstr.put("state", state);

map.put("data", mapstr);

**return** map;

}

/\*\*

\* 查询出的结果明细个数

\*

\* **@param**

\* **@return**

\* **@throws** Exception

\*/

**public** **int** getDetailAllNum(HttpServletRequest request, BsCheckMasterDto dto)

**throws** Exception {

// 使用字符串缓冲器类拼接查询语句

StringBuffer sb = **new** StringBuffer();

sb.append(" bscheckdetail a, omcheckcontent b ");

sb.append(" where a.contentid=b.contentid and a.resultid=:resultid");

sb.append(" and a.detaildecide = :detaildecide ");

**if**(dto.getNum()!=0){

sb.append(" and b.contentimpt =:num");

}

sb.append(" and 1=1 ");

sb.append(" order by a.resultid ");

String sql = sb.toString();

// 转化sql语句

String[] paramName = **new** String[] { "detaildecide","num","resultid" };

**int**[] paramType = **new** **int**[] { Types.*VARCHAR*,Types.*INTEGER*,Types.*VARCHAR* };

sql = QueryFactory.*getSQL*(sql, paramName, paramType, dto);

**int** total = dao.count(sql);

**return** total;

}

String path = Contances.*UPDATERESULTINFO*;

// if (Bimp.tempSelectBitmap.size() > 0) {

saveData(path);

// 安全监管、聚餐申报上传检查结果信息

**public** **static** String *UPDATERESULTINFO* = *baseUrl* + "updateResultinfo";

/\*\*

\*

\* updateResultinfo的中文名称：保存检查结果信息(食品生产企业特殊判断)

\*

\* updateResultinfo的概要说明：穿入reultid和comid comid(需要手机端传入)

\*

\* **@param** dto

\* **@return** Written by : tmm

\*

\*/

@AdaptBy(type = JsonAdaptor.**class**)

@At

@Ok("json")

**public** Object updateResultinfo(HttpServletRequest request, @Param("..") BsCheckMasterDto dto) **throws** Exception {

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

result.put("code", "9999");

result.put("msg", "操作失败");

result.put("success", **false**);

**if** ("null".equalsIgnoreCase(dto.getResultid()) || StringUtils.*isEmpty*(dto.getResultid())) {

result.put("msg", "检查计划ID[resultid]不能为空");

**return** result;

}

**if** ("null".equalsIgnoreCase(dto.getComid()) || StringUtils.*isEmpty*(dto.getComid())) {

result.put("msg", "公司ID[commid]不能为空");

**return** result;

}

**return** SysmanageUtil.*execAjaxResult*(sjbService.updateResultinfo(request, dto));

}

/\*\*

\* 修改结果信息

\*

\*/

**public** String updateResultinfo(HttpServletRequest request,

BsCheckMasterDto dto) {

**try** {

updateResultinfoImpl(request, dto);

} **catch** (Exception e) {

**return** Lang.*wrapThrow*(e).getMessage();

}

**return** **null**;

}

/\*\*

\* 修改结果信息实现方法 使用事务控制

\*

\* **@param** request

\* **@param** dto

\* **@throws** Exception

\*/

@Aop({ "trans" })

**public** **void** updateResultinfoImpl(HttpServletRequest request,

BsCheckMasterDto dto) **throws** Exception {

BsCheckMaster master = dao

.fetch(BsCheckMaster.**class**, dto.getResultid());

**if** (master != **null**) {

PcompanyDTO company =getQiyeInfoByid(request, dto);

// 修改检查完毕标识（4：检查信息固化;5保存完毕）

master.setResultstate(dto.getResultstate());

master.setResultdecision(dto.getResultdecision());

master.setResultng(dto.getResultng());

// master.setResultdate(new Date());// 结束时间

master.setResultperson(dto.getResultperson());

master.setLatitude(dto.getLatitude());

master.setLongitude(dto.getLongitude());

// if(master.getCheckresultinfo()==null || "".equals(master.getCheckresultinfo())){

// //编号

// String number = DbUtils.getOneValue(dao, "select getResultbh("+company.getComdalei()+") from dual");

// dto.setNumber(number);

// }

master.setLxr(dto.getLxr());

master.setLxdh(dto.getLxdh());

**if**("5".equals(dto.getResultstate())){

String result = SysmanageUtil.*getAA027Str*(dto.getAaa027(), dto.getAaa383());

dto.setAaa027(result);

master.setCheckresultinfo(getAssignmentResult(company, dto));

}//告知页赋值陪同人

**if**("1".equals(company.getComdalei())){//食品生产企业

master.setDetailinfo(master.getDetailinfo().replace("ptuser", dto.getResultperson()==**null**?"":dto.getResultperson()));

}**else** {

master.setDetailinfo(master.getDetailinfo().replace("ptuser", " "));

}

**if**(dto.getLocation()==**null** || "".equals(dto.getLocation())){

//获取地理位置

}**else**{

master.setLocation(dto.getLocation());

}

// 修改

dao.update(master);

}

}

/\*\*

\* 结果替换赋值

\* **@param** resultid 计划id

\* **@return**

\* **@throws** Exception

\*/

**public** String getAssignmentResult(PcompanyDTO company,BsCheckMasterDto dto ) **throws** Exception {

String resultinfo ="";

String checkTrue ="☑";

String checkFalse ="□";

String nbsp ="&nbsp; &nbsp;";

**if**(company!=**null**){

dto.setTbodytype("resultInfo");//食品生产经营日常监督检查结果记

dto.setTbodycode("1");

BsTbodyInfoDTo tbodyBean = getTbodyInfo(**null**, dto);

dto.setYear(String.*valueOf*(DateUtil.*convertDateToYear*(**new** Date())));

**int** total = getResultNumBycomid(**null**, dto);//企业本年检查次数

**int** count = getDetailAllNum(**null**, dto);//本次检查明细数

//序号

BsResultNumberDTO result = getresultContent(dto);

//查询

**if**(tbodyBean!=**null**){

//编号

String number = DbUtils.*getOneValue*(dao, "select getResultbh("+company.getComdalei()+") from dual");

String tdodyInfo = tbodyBean.getTbodyinfo();

String starttime = company.getOperatedate();

String resultdisc="";

resultinfo = tdodyInfo.replace("commc", company.getCommc()==**null**?"":company.getCommc())

.replace("commdz", company.getComdz()==**null**?"":company.getComdz())

//.replace("comlxr", company.getComfzr()==null?"":company.getComfzr())

//.replace("comlxfs", company.getComyddh()==null?"":company.getComyddh())

.replace("comlxr", dto.getLxr()==**null**?"":dto.getLxr())

.replace("comlxfs", dto.getLxdh()==**null**?"":dto.getLxdh())

.replace("comxkz", company.getComxkzbh()==**null**?"":company.getComxkzbh())

.replace("year",starttime==**null**?"":starttime.substring(0,4))

.replace("month", starttime==**null**?"":starttime.substring(5,7))

.replace("date",starttime==**null**?"": starttime.substring(8,10))

.replaceFirst("num", String.*valueOf*(total))

.replace("Itemname", company.getItemname()==**null**?"":company.getItemname())

.replaceFirst("count", String.*valueOf*(count))

.replace("fuhe", "101".equals(dto.getResultdecision())?checkTrue:checkFalse)

.replace("bufh","103".equals(dto.getResultdecision())?checkTrue:checkFalse)

.replace("zhenggai", "102".equals(dto.getResultdecision())?checkTrue:checkFalse)

// .replace("resultdisc", dto.getResultng()==null?"":dto.getResultng())

.replaceFirst("jgbh", number==**null**?"":number);

//符号赋值

**if**(result!=**null**){

String impdetail=result.getImpDetail()==**null**?"":result.getImpDetail();

String comdetail=result.getComDetail()==**null**?"":result.getComDetail();

String str ="<p><strong>结果处理：</strong></p><p>"+

("1".equals(dto.getResultFinal())?"☑":"□")+"通过 &nbsp;&nbsp;&nbsp;"+

("2".equals(dto.getResultFinal())?"☑":"□")+"书面限期整改"+

("3".equals(dto.getResultFinal())?"☑":"□")+"食品生产经营者立即停止食品生产经营活动</p>";

System.*out*.println(dto.getResultFinal());

String checkResult=" ";

**if**(!"1".equals(company.getComdalei())){

resultdisc = impdetail+comdetail;

}**else** {//食品生产企业

resultdisc = dto.getResultng();

checkResult=str;

}

resultinfo=resultinfo.replaceFirst("impcount", String.*valueOf*(result.getImpcount()))

.replaceFirst("impnum", result.getImpnum()==**null**?"":result.getImpnum())

.replace("resultdisc", resultdisc==**null**?"":resultdisc)

.replaceFirst("problemcount", String.*valueOf*(result.getProblemcount()))

.replaceFirst("problemnum", result.getProblemnum()==**null**?"":result.getProblemnum())

.replaceFirst("commocount", String.*valueOf*(result.getCommocount()))

.replaceFirst("commnum", result.getCommnum()==**null**?"":result.getCommnum())

.replaceFirst("comproblemcount", String.*valueOf*(result.getComproblemcount()))

.replaceFirst("checkResult", checkResult)

.replaceFirst("comproblemnum", result.getComproblemnum()==**null**?"":result.getComproblemnum());

}

//巡查笔录

dto.setTbodytype("lbxcResultInfo");

dto.setTbodycode("1");

tbodyBean = getTbodyInfo(**null**, dto);

**if**(tbodyBean!=**null**){

tdodyInfo = tbodyBean.getTbodyinfo();

starttime = company.getOperatedate();

String endtime = company.getResultdate();

String lbresultinfo = tdodyInfo.replace("commc", company.getCommc()==**null**?"":company.getCommc())

.replace("comdz", company.getComdz()==**null**?nbsp:company.getComdz())

//.replace("comfrhyz", company.getComfrhyz()==null?nbsp:company.getComfrhyz())

//.replace("comyddh", company.getComyddh()==null?nbsp:company.getComyddh())

.replace("comfrhyz",dto.getLxr()==**null**?nbsp:dto.getLxr())

.replace("comyddh", dto.getLxdh()==**null**?nbsp:dto.getLxdh())

.replace("year",starttime==**null**?nbsp:starttime.substring(0,4))

.replace("month", starttime==**null**?nbsp:starttime.substring(5,7))

.replace("date",starttime==**null**?nbsp: starttime.substring(8,10))

.replaceFirst("time",starttime==**null**?nbsp: starttime.substring(11,13))

.replaceFirst("fen",starttime==**null**?nbsp: starttime.substring(14,16))

.replaceFirst("to\_time",starttime==**null**?nbsp: endtime.substring(11,13))

.replaceFirst("to\_fen",starttime==**null**?nbsp: endtime.substring(14,16));

// .replace("username", "");

resultinfo+=lbresultinfo;

}

}

}**else** {//为其他日常检查

dto.setTbodytype("resultIfo\_jcsb");

dto.setTbodycode("1");

BsTbodyInfoDTo tbodyBean = getTbodyInfo(**null**, dto);

**if**(tbodyBean!=**null**){

String tdodyInfo = tbodyBean.getTbodyinfo();

resultinfo = tdodyInfo.replace("Itemname", "农村聚餐日常监督检查")

.replace("fuhe", "101".equals(dto.getResultdecision())?checkTrue:checkFalse)

.replace("bufh","103".equals(dto.getResultdecision())?checkTrue:checkFalse)

.replace("zhenggai", "102".equals(dto.getResultdecision())?checkTrue:checkFalse)

.replace("resultdisc", dto.getResultng()==**null**?"":dto.getResultng());

}

}

**return** resultinfo;

}

/\*\*

\* 查询企业年度检查次数

\*

\* **@param**

\* **@return**

\* **@throws** Exception

\*/

**public** **int** getResultNumBycomid(HttpServletRequest request, BsCheckMasterDto dto)

**throws** Exception {

// 使用字符串缓冲器类拼接查询语句

StringBuffer sb = **new** StringBuffer();

// sb.append(" bscheckmaster a , bscheckplan b ");

// sb.append(" where a.planid = b.planid and b.planchecktype =:planchecktype ");

sb.append(" bscheckmaster a ");

sb.append(" where a.comid=:comid and a.operatedate like :year ");

sb.append(" and 1=1 ");

sb.append(" order by a.resultid ");

String sql = sb.toString();

// 转化sql语句

String[] paramName = **new** String[] { "comid","year" };

**int**[] paramType = **new** **int**[] { Types.*VARCHAR*,Types.*VARCHAR* };

sql = QueryFactory.*getSQL*(sql, paramName, paramType, dto);

**int** total = dao.count(sql);

**return** total;

}

/\*\*

\* 查询结果序号内容

\* **@param** resultid 计划id

\* **@return**

\* **@throws** Exception

\*/

**public** BsResultNumberDTO getresultContent(BsCheckMasterDto dto) **throws** Exception {

//重要性

BsCheckMasterDto master = **new** BsCheckMasterDto();

BsResultNumberDTO result = **new** BsResultNumberDTO();

//总数

dto.setNum(1);

master = getImpDetailNumber(**null**, dto);

**if**(master!=**null**){

result.setImpcount(master.getImpnum());

result.setImpnum(master.getStr());

}

//不及格数（重要性）

dto.setDetaildecide("2");

master = getImpDetailNumber(**null**, dto);

**if**(master!=**null**){

result.setProblemcount(master.getImpnum());

result.setProblemnum(master.getStr());

result.setImpDetail(master.getImpDetail());

}

//一般性

dto.setDetaildecide(**null**);

master = getGeneralDetailNumber(**null**, dto);

**if**(master!=**null**){

result.setCommocount(master.getImpnum());

result.setCommnum(master.getStr());

}

//不及格数（一般性）

dto.setDetaildecide("2");

master = getGeneralDetailNumber(**null**, dto);

**if**(master!=**null**){

result.setComproblemcount(master.getImpnum());

result.setComproblemnum(master.getStr());

result.setComDetail(master.getComDetail());

}

**return** result;

}

/\*\*

\* （重要项）查询具体编号和个数(num 1为不设置)

\*

\* **@param**

\* **@return**

\* **@throws** Exception

\*/

**public** BsCheckMasterDto getImpDetailNumber(HttpServletRequest request, BsCheckMasterDto dto)

**throws** Exception {

// 使用字符串缓冲器类拼接查询语句

StringBuffer sb = **new** StringBuffer();

sb.append(" select d.contentcode ");

**if**("2".equals(dto.getDetaildecide())){

sb.append(" ,CONCAT(d.contentcode,'.',d.content) as content ");

}

sb.append(" from bscheckmaster a ,bscheckdetail c ,omcheckcontent d ");

sb.append(" where c.resultid = a.resultid and d.contentid = c.contentid ");

sb.append(" and c.detaildecide =:detaildecide and a.resultid =:resultid ");

sb.append(" and d.contentimpt <>:num");

sb.append(" and 1=1 ");

sb.append(" order by d.contentsortid ");

String sql = sb.toString();

// 转化sql语句

String[] paramName = **new** String[] { "detaildecide","resultid","num"};

**int**[] paramType = **new** **int**[] { Types.*VARCHAR*,Types.*VARCHAR*,Types.*INTEGER*};

sql = QueryFactory.*getSQL*(sql, paramName, paramType, dto);

//查询

Map map = DbUtils.*DataQuery*(GlobalNames.*sql*, sql, **null**, BsCheckMasterDto.**class**);

List<BsCheckMasterDto> ls = (List<BsCheckMasterDto>) map.get(GlobalNames.*SI\_RESULTSET*);

**if**(ls.size()>0){

StringBuffer str =**new** StringBuffer();

StringBuffer contentstr =**new** StringBuffer();

BsCheckMasterDto masterBean = **new** BsCheckMasterDto();

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

//数据太长是表格变形，所以每8个换一次行

**if**(i%10==0&&i!=0){

str.append("</br>");

}

str.append(ls.get(i).getContentcode()).append(",");

**if**(ls.get(i).getContent()!=**null**&&!"".equals(ls.get(i).getContent())){

contentstr.append(ls.get(i).getContent()).append(" </br>");

}

}

**if**(contentstr.toString()!=**null**&&!"".equals(contentstr.toString())){

masterBean.setImpDetail(contentstr==**null**?"":contentstr.toString());

}

masterBean.setStr(str.substring(0, str.lastIndexOf(",")).toString());

masterBean.setImpnum(ls.size());

**return** masterBean;

}

**return** **null**;

}

/\*\*

\* （一般项）查询具体编号和个数(num 1为不设置)

\*

\* **@param**

\* **@return**

\* **@throws** Exception

\*/

**public** BsCheckMasterDto getGeneralDetailNumber(HttpServletRequest request, BsCheckMasterDto dto)

**throws** Exception {

// 使用字符串缓冲器类拼接查询语句

StringBuffer sb = **new** StringBuffer();

sb.append(" select d.contentcode ");

**if**("2".equals(dto.getDetaildecide())){

sb.append(" ,CONCAT(d.contentcode,'.',d.content) as content ");

}

sb.append(" from bscheckmaster a ,bscheckdetail c ,omcheckcontent d ");

sb.append(" where c.resultid = a.resultid and d.contentid = c.contentid ");

sb.append(" and c.detaildecide =:detaildecide and a.resultid =:resultid ");

sb.append(" and d.contentimpt =:num");

sb.append(" and 1=1 ");

sb.append(" order by d.contentsortid ");

String sql = sb.toString();

// 转化sql语句

String[] paramName = **new** String[] { "detaildecide","resultid","num"};

**int**[] paramType = **new** **int**[] { Types.*VARCHAR*,Types.*VARCHAR*,Types.*INTEGER*};

sql = QueryFactory.*getSQL*(sql, paramName, paramType, dto);

Map map = DbUtils.*DataQuery*(GlobalNames.*sql*, sql, **null**, BsCheckMasterDto.**class**);

List<BsCheckMasterDto> ls = (List<BsCheckMasterDto>) map.get(GlobalNames.*SI\_RESULTSET*);

**if**(ls.size()>0){

StringBuffer str =**new** StringBuffer();

StringBuffer contentstr =**new** StringBuffer();

BsCheckMasterDto masterBean = **new** BsCheckMasterDto();

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

//数据太长是表格变形，所以每8个换一次行

**if**(i%10==0&&i!=0){

str.append("</br>");

}

str.append(ls.get(i).getContentcode()).append(",");

**if**(ls.get(i).getContent()!=**null**&&!"".equals(ls.get(i).getContent())){

contentstr.append(ls.get(i).getContent()).append(" </br>");

}

}

**try** {

// System.out.println("\*\*\*\*\*\*\*\*\*\*\*\*"+contentstr.toString());

**if**(contentstr.toString()!=**null**&&!"".equals(contentstr.toString())){

masterBean.setComDetail(contentstr.toString());

}

} **catch** (Exception e) {

e.printStackTrace();

}

masterBean.setStr(str.substring(0, str.lastIndexOf(",")).toString());

masterBean.setImpnum(ls.size());

**return** masterBean;

}

**return** **null**;

}

保存

**public** **void** onClick(View view) {

check\_opinion = et\_input\_check\_opinion.getText().toString().trim();

orgnaization\_accompanying\_person = et\_orgnaization\_accompanying\_person.getText().toString().trim();

result\_location = tv\_result\_location.getText().toString().trim();

**switch** (view.getId()) {

**case** R.id.*but\_save\_data*:

**if** (bt\_match\_provisions.isChecked() == **false** && bt\_unmatch\_provisions.isChecked() == **false**

&& bt\_limit\_date\_rectification.isChecked() == **false**) {

showToast("检查结果为空...");

**return**;

}

**if** (**null** == pDialog) {

pDialog = **new** ProgressDialog(ctx);

}

pDialog.setMessage("正在保存，请稍后...");// 设置对话框中的提示信息

pDialog.setCancelable(**false**);

pDialog.show();

**try** {

params.put("aaa027", aaa027);

params.put("aaa383", aaa383);

params.put("resultid", resultid);

// 结果不符合说明

params.put("resultng", check\_opinion);

// 结果判定

params.put("resultdecision", index);

// 结果处理

params.put("resultfinal", result\_index);

// 保存或是提交的标识

params.put("resultstate", 5);

params.put("comid", comid);

params.put("location", result\_location);

params.put("latitude", latitude);

params.put("longitude", longitude);

params.put("resultperson", orgnaization\_accompanying\_person);

params.put("lxr", et\_enterprise\_responsible\_person\_name.getText().toString());

params.put("lxdh", et\_enterprise\_responsible\_person\_mobile.getText().toString());

} **catch** (JSONException e) {

e.printStackTrace();

} **catch** (Exception e) {

e.printStackTrace();

showToast("保存失败，请重新保存...");

}

String path = Contances.*UPDATERESULTINFO*;

// if (Bimp.tempSelectBitmap.size() > 0) {

saveData(path);

// } else {

// showToast("请添加附件");

// pDialog.dismiss();

// return;

// }

**break**;

**case** R.id.*iv\_back*:

finish();

**break**;

**default**:

**break**;

}

}

// 安全监管、聚餐申报上传检查结果信息

**public** **static** String *UPDATERESULTINFO* = *baseUrl* + "updateResultinfo";

提交按钮

**private** **void** showUpdateDialog() {

AlertDialog.Builder adb = **new** AlertDialog.Builder(CheckResultInformationActivity.**this**);

adb.setTitle("保存成功");

adb.setMessage("亲，已经保存成功，是否要提交数据，提交后将无法再改变数据！");

adb.setCancelable(**false**);

**final** String check\_opinion = et\_input\_check\_opinion.getText().toString();

adb.setPositiveButton("确定", **new** DialogInterface.OnClickListener() {

@Override

**public** **void** onClick(DialogInterface dialog, **int** which) {

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

dialog.dismiss();

**if** (**null** == pDialog) {

pDialog = **new** ProgressDialog(ctx);

}

pDialog.setMessage("正在提交，请稍后...");// 设置对话框中的提示信息

pDialog.show();

**try** {

params.put("aaa027", aaa027);

params.put("aaa383", aaa383);

params.put("resultid", resultid);

// 结果不符合说明

params.put("resultng", check\_opinion);

// 结果判定

params.put("resultdecision", index);

// 结果处理

params.put("resultfinal", result\_index);

// 保存或是提交的标识

params.put("resultstate", 4);

params.put("comid", comid);

} **catch** (Exception e) {

e.printStackTrace();

}

**final** String commitPath = Contances.*UPDATERESULTINFO*;

**new** Thread() {

**public** **void** run() {

**try** {

URL url = **new** URL(commitPath);

String content = String.*valueOf*(params);

/\* 把JSON数据转换成String类型使用输出流向服务器写 \*/

System.*out*.println("///////////////////" + content);

HttpURLConnection conn = (HttpURLConnection) url.openConnection();

conn.setRequestMethod("POST");

conn.setConnectTimeout(60000);

conn.setRequestProperty("User-Agent", "Fiddler");

conn.setRequestProperty("Content-Type", "application/json");

conn.setRequestProperty("Charset", "utf-8");

OutputStream os = conn.getOutputStream();

os.write(content.getBytes());

os.close();

/\* 服务器返回的响应码 \*/

**int** code = conn.getResponseCode();

**if** (code == 200) {

BufferedReader in = **new** BufferedReader(

**new** InputStreamReader(conn.getInputStream(), "utf-8"));

String retData = **null**;

String responseData = "";

**while** ((retData = in.readLine()) != **null**) {

responseData += retData;

}

System.*out*.println(responseData + "================check");

JSONObject jsonObject = **new** JSONObject(responseData);

String codeData = jsonObject.getString("code");

// **TODO** 如果更新代码，需要改0或者0000，版本更新时候进行测试更改

**if** (codeData.equals("0") || codeData.equals("0000")) {

runOnUiThread(**new** Runnable() {

**public** **void** run() {

pDialog.dismiss();

Bimp.*tempSelectBitmap*.clear();

Bimp.*max* = 0;

showToast("提交成功");

Intent intent = **new** Intent(ctx, ShowResultActivity.**class**);

intent.putExtra("resultid", resultid);

System.*out*.println("@-------===========3333=-@" + resultid);

startActivity(intent);

finish();

}

});

}

in.close();

} **else** {

pDialog.dismiss();

Toast.*makeText*(getApplicationContext(), "提交失败", Toast.*LENGTH\_SHORT*).show();

}

} **catch** (Exception e) {

pDialog.dismiss();

}

}

}.start();

}

});

提交成功后进入打印页面

package com.askj.tangyin.drug.app;

public class ShowResultActivity extends Activity implements OnClickListener {

// 核查意见及结果的接口

**public** **static** String *showResultPrintUrl* = *baseUrl* + "getResultHtmlByresultId";

resultid = intent.getStringExtra("resultid");

path = Contances.*showResultPrintUrl* + "?resultid=" + resultid;

webView = (WebView) findViewById(R.id.*webView*);

webView.getSettings().setJavaScriptEnabled(**true**);

// 设置可以支持缩放

webView.getSettings().setSupportZoom(**true**);

// 设置出现缩放工具

webView.getSettings().setBuiltInZoomControls(**true**);

// 扩大比例的缩放

webView.getSettings().setUseWideViewPort(**true**);

// 自适应屏幕

webView.getSettings().setLayoutAlgorithm(LayoutAlgorithm.*SINGLE\_COLUMN*);

webView.getSettings().setLoadWithOverviewMode(**true**);

**if**(resultid!=**null**){

**if**(resultid.length()<26){

webView.loadUrl(path);

}**else**{

webView.loadUrl(resultid+"&userid="+sp.getString("userid", ""));

bt\_print.setVisibility(View.*GONE*);

//可以在webView中打开网页

webView.setWebViewClient(**new** WebViewClient(){

@Override

**public** **boolean** shouldOverrideUrlLoading(WebView view, String url) {

resultidss=url;

**if**(url.contains("getajdjDocumentsHtml")){

view.loadUrl(url);

bt\_print.setVisibility(View.*VISIBLE*);

}

**if**(url.contains("getflfg")){

// 选择案件发生理由

Intent intent = **new** Intent(ctx, ChooseCaseReasonActivity.**class**);

startActivityForResult(intent, 1004);

System.*out*.println("################"+"当场行政处罚决定书第一个选择按钮");

}

**if**(url.contains("getyjflfg")){

//访问当场行政处罚书第二个按钮

// http://192.168.191.1:8080/syjzhpt/common/sjb/queryPcyzdsz?tabname=zfwsdcxzcfjds29&colname=dcxzcfjdsyjflfg

// 选择依据法律法规

Intent intent = **new** Intent(ctx, ChooseFineReasonActivity.**class**);

intent.putExtra("flfg", "getyjflfg");

startActivityForResult(intent, 1003);

System.*out*.println("################"+"当场行政处罚书第二个按钮");

}

**if**(url.contains("get3pcyzdsz")){

// 责令整改通知书

//http://192.168.191.1:8080/syjzhpt/common/sjb/queryPcyzdsz?tabname=zfwszlgztzs20&colname=zlgzwfgd

Intent intent = **new** Intent(ctx, ChooseFineReasonActivity.**class**);

intent.putExtra("flfg", "get3pcyzdsz");

startActivityForResult(intent, 1003);

System.*out*.println("################"+"访问责令整改通知书第一个按钮");

}

**if**(url.contains("getpcyzdsz")){

// 责令整改通知书

//http://192.168.191.1:8080/syjzhpt/common/sjb/queryPcyzdsz?tabname=zfwszlgztzs20&colname=zlgzcfgj

Intent intent = **new** Intent(ctx, ChooseFineReasonActivity.**class**);

intent.putExtra("flfg", "getpcyzdsz");

startActivityForResult(intent, 1004);

System.*out*.println("################"+"访问责令整改通知书第二个按钮");

}

**return** **true**;

}

});

}

}

/\*\*

\*

\* getResultHtmlByresultId的中文名称：根据结果id获取结果明细页面信息

\*

\* getResultHtmlByresultId的概要说明：传入resultid

\*

\* **@param** dto

\* **@return** Written by : tmm

\*

\*/

@At

@Ok("raw:htm")

**public** Object getResultHtmlByresultId(HttpServletRequest request, @Param("..") BsCheckMasterDto dto) {

// Map map = new HashMap();

// try {

// map.put("data", sjbService.getResultDetail(request,

// dto).getDetailinfo());

// return SysmanageUtil.execAjaxResult(map, null);

// } catch (Exception e) {

// return SysmanageUtil.execAjaxResult(map, e);

// }

**try** {

**return** sjbService.getResultDetail(request, dto).getDetailinfo();

} **catch** (Exception e) {

**return** "暂无数据";

}

}

/\*\*

\* 查询出结果对象

\*

\* **@param**

\* **@return**

\* **@throws** Exception

\*/

**public** BsCheckMaster getResultDetail(HttpServletRequest request,

BsCheckMasterDto dto) **throws** Exception {

BsCheckMaster master = dao

.fetch(BsCheckMaster.**class**, dto.getResultid());

StringBuffer sb = **new** StringBuffer();

sb.append("<!DOCTYPE html><html> <head><title>结果明细</title><meta http-equiv='content-type' content='text/html; charset=UTF-8'>");

sb.append("<meta name='Author' content=''> <meta name='Keywords' content=''> <meta name='Description' content=''>");

sb.append("</head><body>");

**if** (master != **null**) {

Bscheckplan plan = dao

.fetch(Bscheckplan.**class**, master.getPlanid());

// gu20161114 Pcompany pcompany = dao

// .fetch(Pcompany.class, master.getComid());

**if**(plan!=**null**){

String type = plan.getPlanchecktype();

**if**("0".equals(type)){

//量化明细表

// sb.append(master.getCheckresultinfo());

sb.append(master.getDetailinfo());

}**else** **if**("1".equals(master.getCheckgroupstate())) {

//日常结果记录表

sb.append(master.getDetailinfo());

}**else** {

//日常结果记录表

sb.append(master.getCheckresultinfo());

sb.append(master.getDetailinfo());

// if(pcompany!=null){//企业（食品生产显示明细）

// if("1".equals(pcompany.getComdalei())){//食品生产

// sb.append(master.getDetailinfo());

// }

// }

}

}

} **else** {

master = **new** BsCheckMaster();

sb.append("暂无数据");

}

sb.append("</body></html>");

master.setDetailinfo(sb.toString());

**return** master;

}

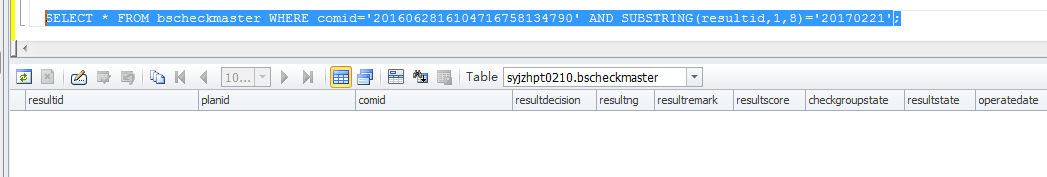
6：涉及到的流程图

7：跳转到的页面

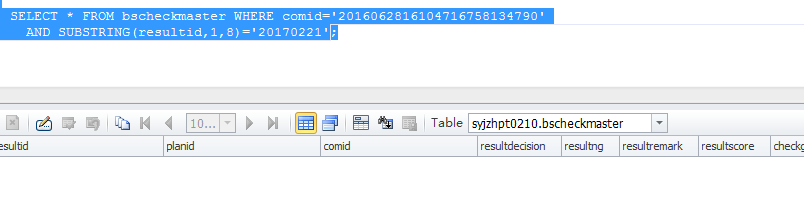
Comid=2016062816104716758134790



SELECT \* FROM bscheckmaster WHERE comid='2016062816104716758134790' AND SUBSTRING(resultid,1,8)='20170221';

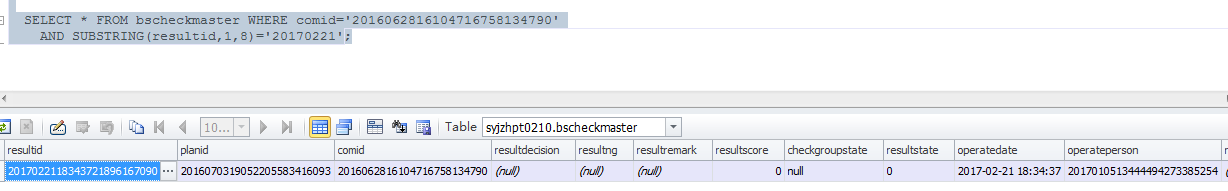


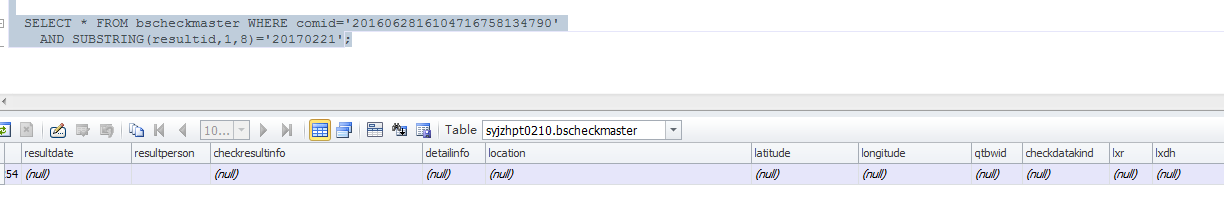


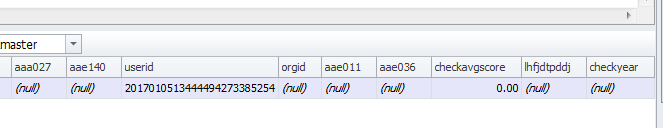




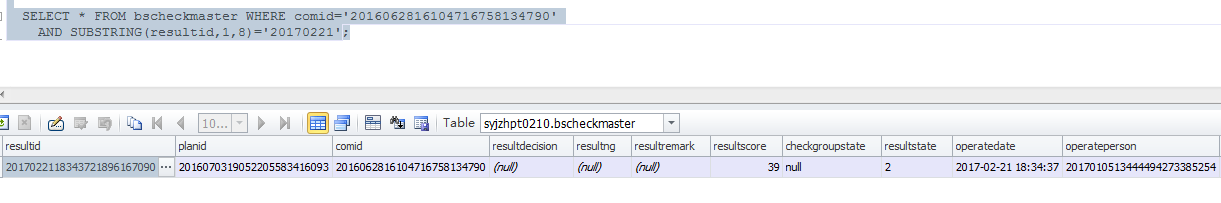
未点保存

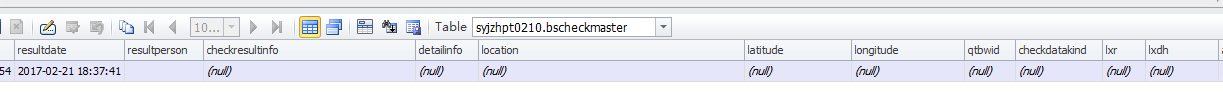


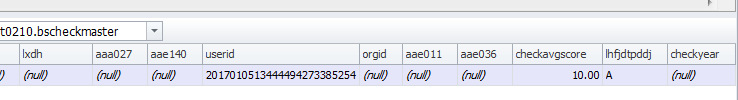




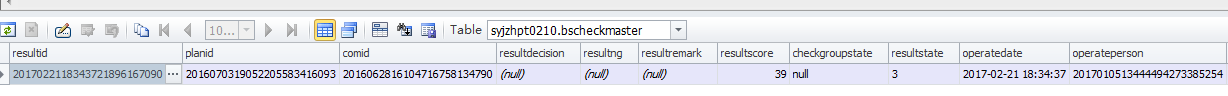
保存后未提交

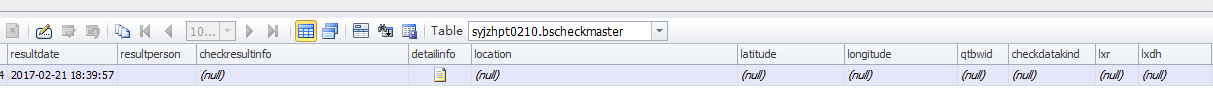


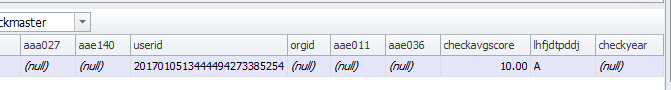


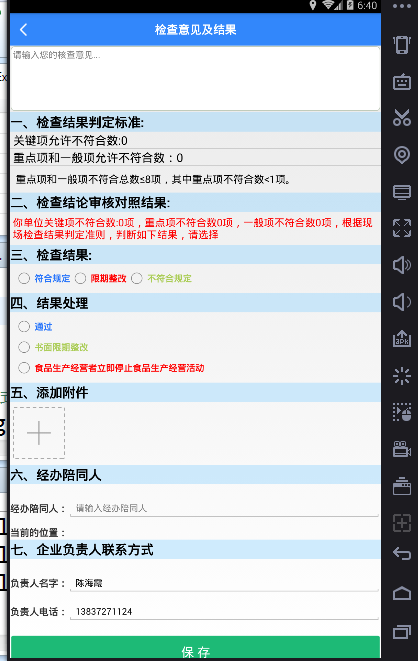


提交后

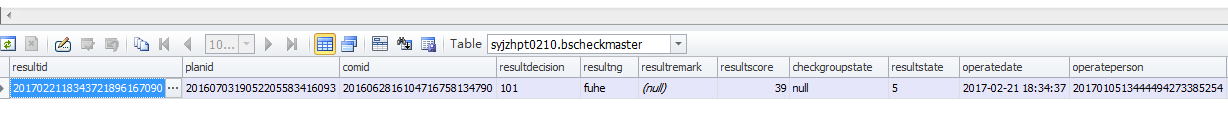


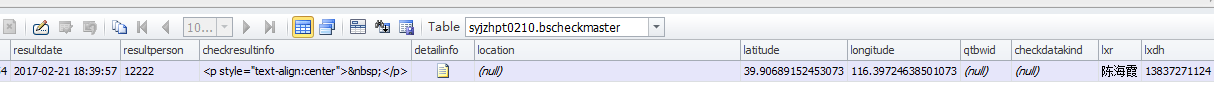


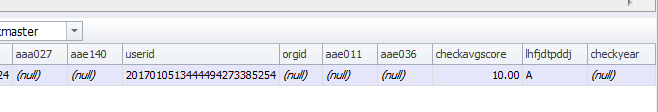


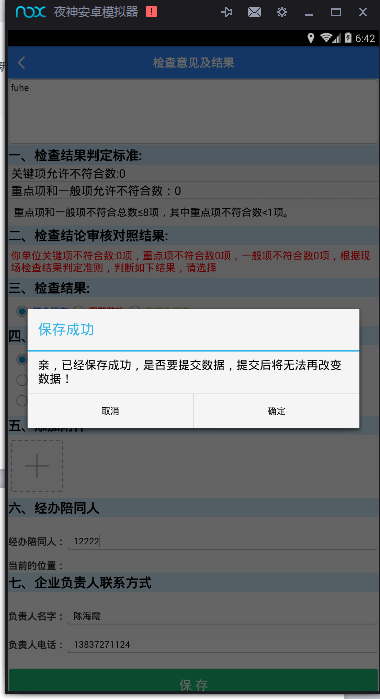


保存后









确定后

