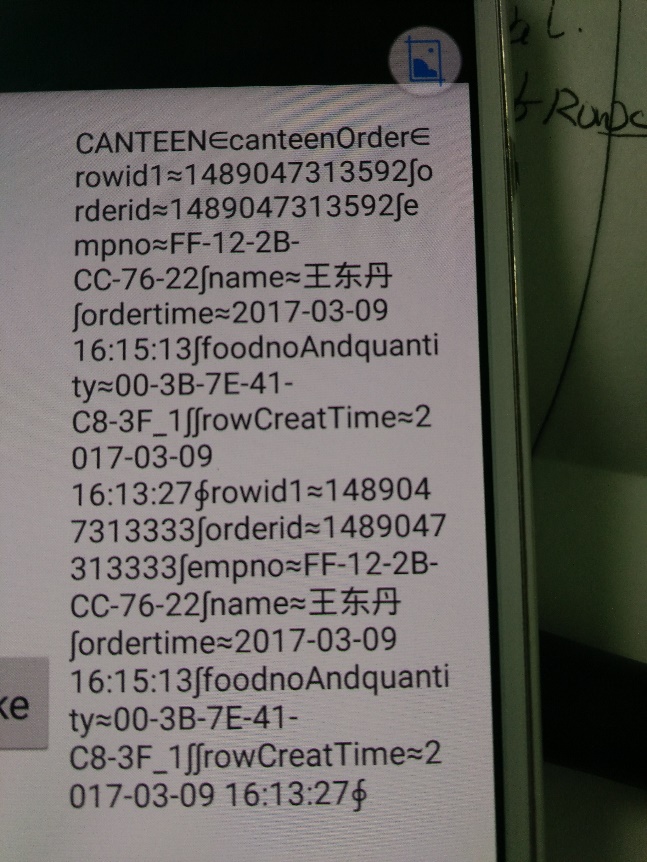
**范例：**

**（1）//查询多条数据**

case R.id.task\_searchRows:

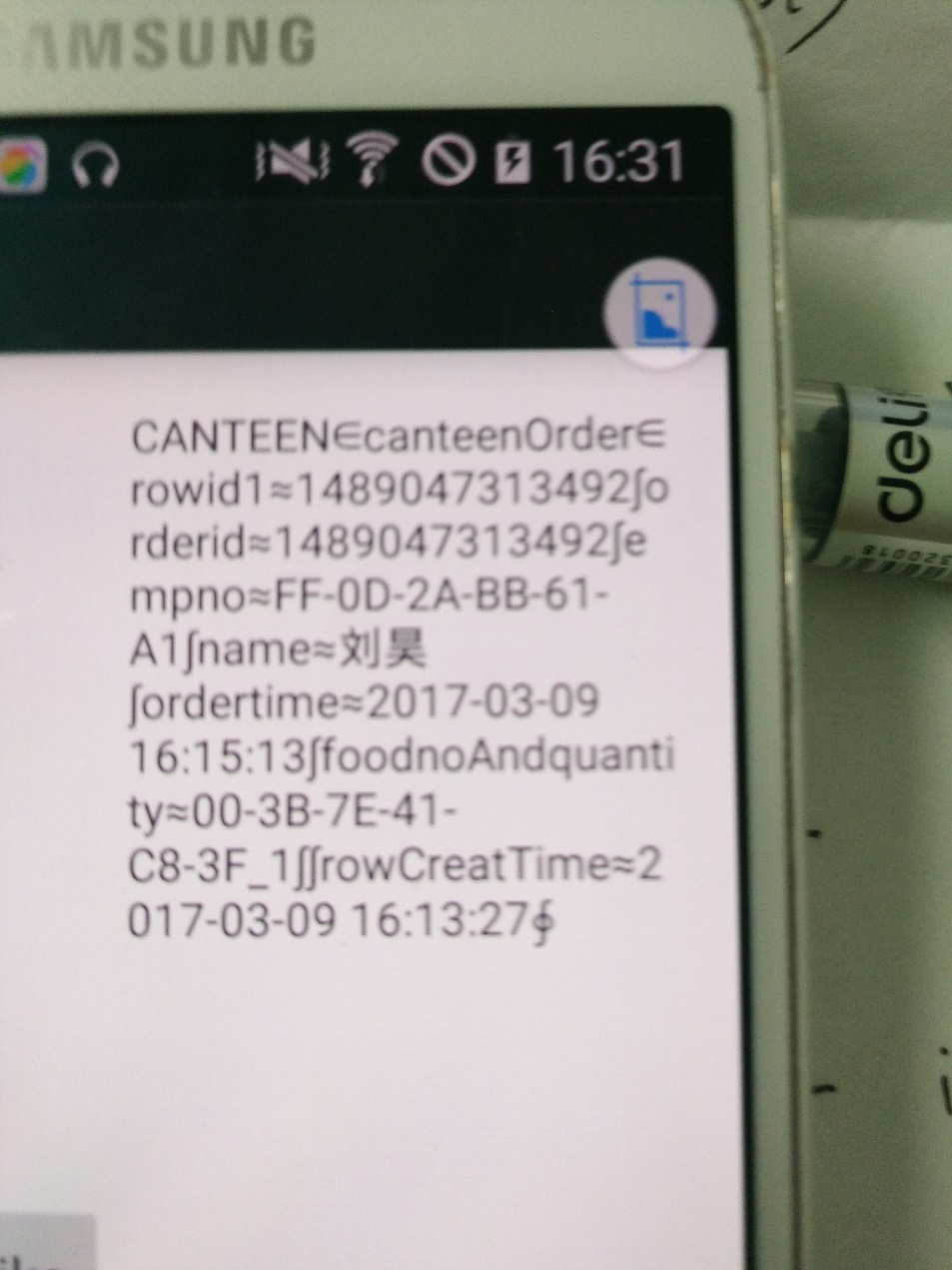
serchRows(InfoType.SearchRows,"CANTEEN"+Separator.at\_Str+"canteenOrder"+Separator.at\_Str+"empno"+Separator.Sign+MyApplication.mac,PathType.unit);break;



**（2）//模糊查询**

case R.id.task\_searchRowsLike:

serchRows(InfoType.SearchRowsLike,"CANTEEN"+Separator.at\_Str+"canteenOrder"+Separator.at\_Str + "foodnoAndquantity" + Separator.Sign + "00-3B-7E-41-C8-3F\_1",PathType.unit);

 break;

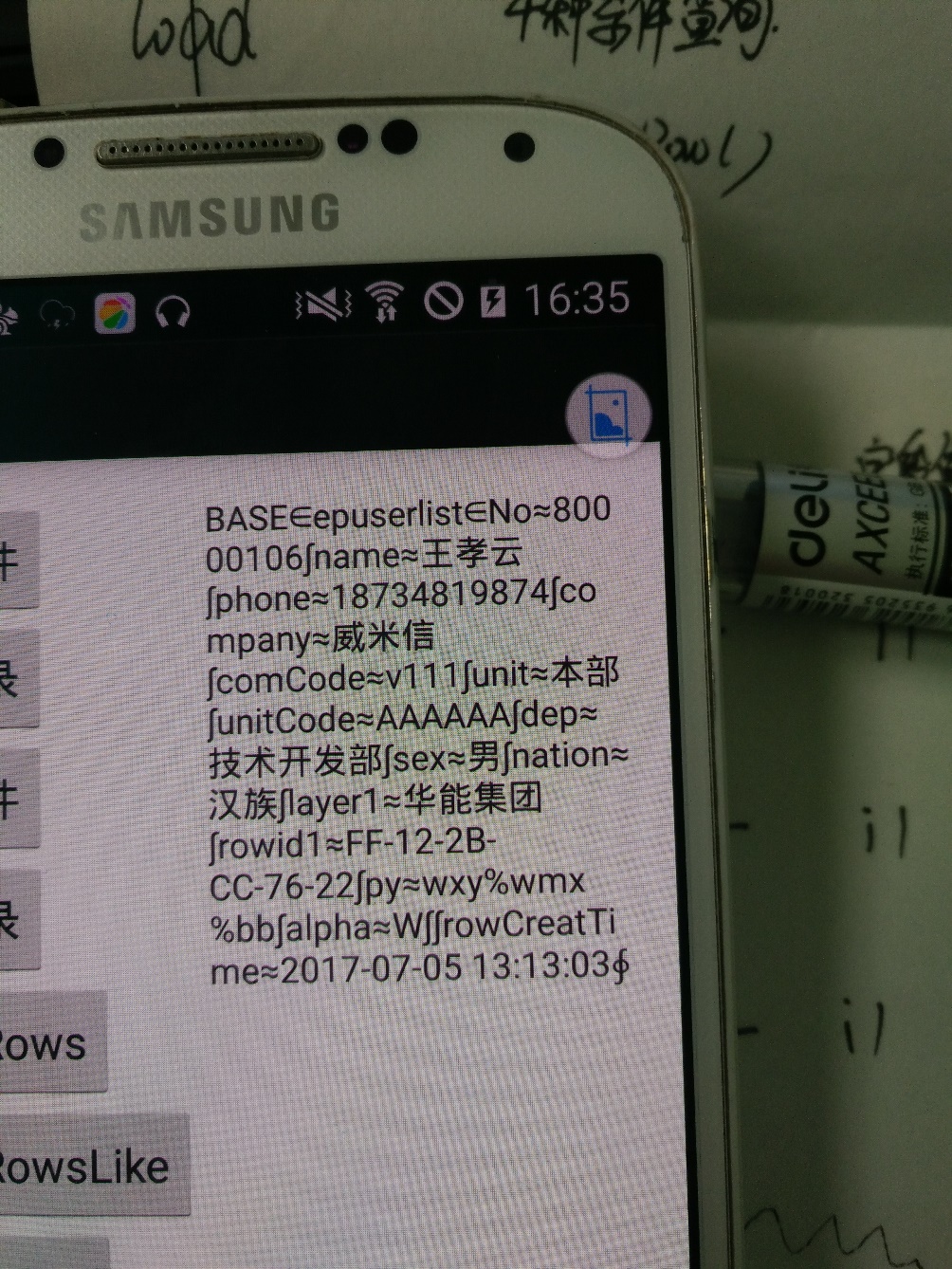
**（3）//通过Id获取一条数据**

case R.id.task\_GetRowById:

String content = "BASE" + Separator.at\_Str + "epuserlist" + Separator.at\_Str + MyApplication.mac;

serchRows(InfoType.GetRowById,content, PathType.pub);

break;



**（5）//增加或者修改**

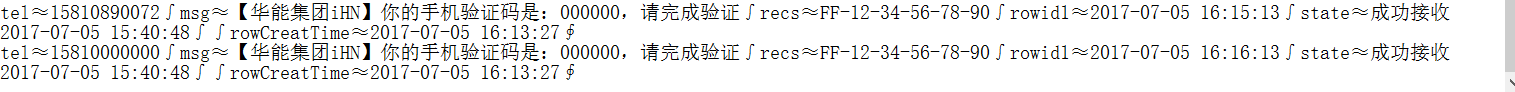
case R.id.task\_updateById:

String rowStr = "tel≈15899999999∫msg≈【华能集团iHN】你的手机验证码是：999999，请完成验证∫recs≈FF-12-34-56-78-90∫" +"rowid1≈2017-07-05 16:18:13∫state≈成功接收 2017-07-05 15:40:48" +"∫∫rowCreatTime≈2017-07-05 16:13:27";

**//上传到服务器**

sendRowServer(InfoType.updateById,PathType.pub,"SMSSET"+Separator.at\_Str + "sendtbl"+ Separator.at\_Str + rowStr);

break;



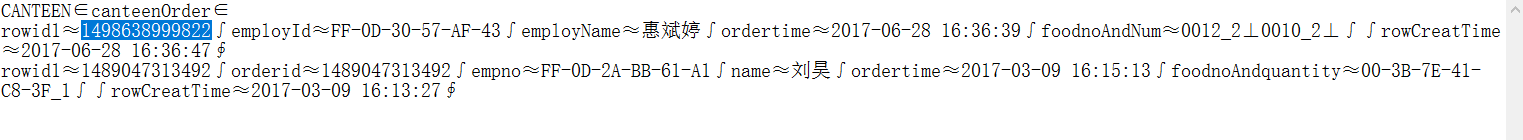
**（6）//通过Id删除一条数据**

**case R.id.task\_delById:**

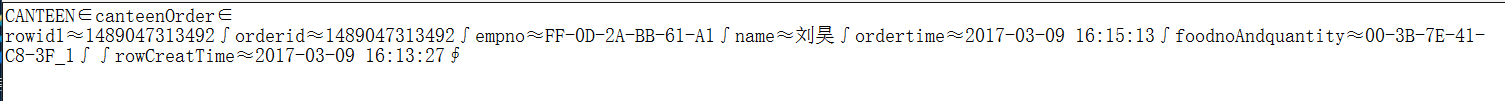
sendRowServer(InfoType.delById, PathType.unit, "CANTEEN" + Separator.at\_Str + "canteenOrder"+ Separator.at\_Str + "1489047317649");

break;

执行操作之前：



执行操作之后：



**例子**：当点击查询时，在ListView中显示两个字段值且tel字段没有重复值，在TextView中显示总数。

private void **searchRows()** {

serchRows(InfoType.SearchRowsLike, "SMSSET" + Separator.at\_Str + "sendtbl"+ Separator.at\_Str + "state" + Separator.Sign + "成功接收",PathType.pub);

}

private void **serchRows**(InfoType infoType, String condition, PathType pathType) {

GetDataInfo getDataInfo = new GetDataInfo(condition,

infoType, pathType, MyApplication.socketObj.getIp(),

MyApplication.socketObj.getPort(),

MyApplication.socketObj,

new ResultStrCallBack() {

@Override

public void onResult(String result) {

Tool.showLog("成功返回值", "result " + result);

getRows(result);

show\_tv.setText("成功接收为：" + num + "个");

}

});

SocketUtil.getData(getDataInfo);

}

public ArrayList<Row> **getRows**(String rowStrs) {

ArrayList<Row> rows = new ArrayList<Row>();

if (TextUtils.isEmpty(rowStrs)) return rows;

if (rowStrs.split("∈").length <= 2) {

return rows;

}

String temp=rowStrs.split("∈")[2];

String[] perRowStr = temp.split(Separator.semicolon);

//用HashMap过滤重复值

HashMap<String, Row> map = new HashMap<>();

for (String str : perRowStr) {

if (TextUtils.isEmpty(str)) continue;

Row row = new Row(str);

row.rowStr = str;

rows.add(row);

//键值对，key为唯一值，value为一条数据

map.put(row.getVal("tel"), row);

}

//key只能处理成set对象，value只能处理成Colletion对象

//map提供给一个方法keySet()来得到所有key的集合

java.util.Set<String> set = map.keySet();

ArrayList<Row> rowsList = new ArrayList<>();

for (String key : set) {

rowsList.add(map.get(key));

}

num=rowsList.size();

mItemLayout = R.layout.list\_item;

mAdapter = new Adapter(Wangdongdan.this, mItemLayout, rowsList);

lvs.setAdapter(mAdapter);

return rows;

切记for循环语句，

