4g主动定位项目注意事项

1. 网络层

负责处理与服务器的网络数据交互

1.1 com.position4g.service.SocketService

后台服务,负责管理socket与线程

1.1.1 init()方法初始化socket连接,mSocket1连接管理服务器,端口号16385,mSocket2连接上报服务器,端口号16387

Socket断连时重新调用此方法初始化连接

1.1.2 PowerManager.WakeLock保持cpu唤醒,后台不被杀死

1.1.3 timeRunnable定时器,每5秒发送心跳包

心跳包接口:{"stats":4128, "timestamp": 1427247078}

1.2 com.position4g.thread.ParamThread1 连接管理服务器的线程

1.2.1 byte bf[] = new byte[10\*1024]每次读取了10\*1024字节数组保证数据完整(数据量大时4K读不完整)

1.2.2 EventBus负责所有页面之间的数据传递,标注有(@Subscribe)的方法将在数据传递时被调用

1.2.3 三次握手

第一次握手

{“Welecome to tcp server!^M”}

第一次握手返回信息

{ "devNumber": "SJLTE0006", "devName": "GuangZhouLTE", "typeModel": "WDG10105", "phoneNumber": "19925356", "devPos": { "height": 3.500000, "longitude": 0.010000, "latitude": 0.010000 }, "devAddresss": "guangzhoukeyun", "mac": "00:E2:69:03:E5:17", "paMaxPower": 0, "devType": 9, "devConformation": 1, "timestamp": 1490079937 }

第二次握手

{"code":54,"msg":"Device registered successfully.","stats":4129,"timestamp":1490071782}

1.3 com.position4g.thread.ParamThread2 连接上报服务器的线程,同上

1. 工具类

2.1 com.position4g.utils.DataUtilsjson数据处理类,解析json 字符串为javabean对象

2.1.1 getLen(String data)获取第一段完整json的长度,防止粘包

2.1.2 没有在方法里开子线程,若调用数据库在调用时开启子线程

2.2 com.position4g.utils.PreferenceUtils sharedpreference的全局管理类

2.2.1 sharedpreference数据在卸载后自动清空,直接覆盖安装不清空

2.2.2 clearData()方法可以手动清空sharedpreference保存的所有数据(暂时没用到)

2.3 com.position4g.utils.ThreadUtils 线程池管理类

2.3.1 ThreadUtils.getThreadPoolProxy().execute(new runnable())从线程池获取一个子线程并加入队列

2.3.2 ThreadUtils.runOnMainThread(new runnable())强制在UI线程运行

2.4 com.position4g.utils.ToastUtil toast的管理类,简化代码,防止连续点击时出现toast长期占屏的情况

1. UI界面

3.1 com.position4g.ui.activity.HomeActivity 首页根布局

3.1.1 initAddressDb()默认进行检测,如果本地不存在address.db(省市数据库),则会从资源文件中拷贝一份至本地

3.1.2 onDevTypeMessage初始化页面时判断当前状态为电信/联通.这个数据在第一次握手时传过来,如果想放在设备fragment中则需要设置 viewpager.setOffscreenPageLimit(2)

3.1.3 重写返回键事件onBackPressed() moveTaskToBack(false)点击将APP移至后台

3.1.4 顶端移除actionbar,使用toolbar作为标题栏,根据左侧

drawerlayout中listview被点击的position更改标题内容

3.2 com.position4g.ui.fragment.Target\_HomeFragment 目标记录页面

目标记录列表数据接口:

{ "imsi": "460000560736203", "delay": 11, "sinr": 12, "freq": 39098, "bbu": 1, "rsrp": 12 }

Javabean:com.position4g.model.TargetBean

3.2.1柱状图控件: compile 'com.github.PhilJay:MPAndroidChart:v3.0.2'

3.2.1.1柱状图barchat动态添加数据会导致X轴缩放,所以每次新增数据都对柱状图设置进行初始化data.groupBars(0.3f, 0.25f, 0.05f)

3.2.1.2 barchat没有提供动态更新数据api,设定X轴显示最大数值(8),超过时删除第一个数据并刷新视图,伪装数据动态刷新并左移的效果

3.2.1.3 PreferenceUtils.getBoolean(Constants.SHOWDELAY, true)为true时显示时延,否则不显示,在设置页面切换选项

3.2.1.4 xAxis.setAxisMinimum(-0.5f);把第一条数据坐标轴右移一点,防止左侧被覆盖

3.2.1.5 如果有其他需求难以实现建议换成折线图LineChart

3.2.2 codeDatas为帧码记录页面列表数据集合,已移至帧码记录页面处理

3.2.3 mDatas为目标记录页面列表数据集合,同一imsi只显示一条数据,有新的数据时覆盖之前的,不新增相同imsi的item

3.2.4 点击item选中当前imsi,如果没有选择默认listview中第一条为选中状态,柱状图imsi与listview同步

3.2.5 音频文件不全,取值时向下取值一直到有对应的文件

3.2.5 getWindow().addFlags(WindowManager.LayoutParams.FLAG\_KEEP\_SCREEN\_ON);保持屏幕在app运行时常量,不会自动息屏

3.3 com.position4g.ui.fragment.Code\_HomeFragment帧码记录页面

帧码记录列表接口(从上报服务器获取):[{"cellNumber":2,imei":"486544563372875","imsi":"460020286433452260","tmsi":"","uptime":1426014675 }]

3.3.1 ‘总量’记录所有接收到数据(实际可显示的数据不超过10000条),’编号’根据不同imsi出现先后次序赋予,相同imsi对应一个编号

3.3.2 每接受到一条数据就向上报服务器返回一条回复

{"upresult":true,"srcTime":1301236456,"time":13012346}

3.3.3 sumRunnable每60秒刷新一次数据,统计接收到的数据(每分钟上号)加入总量,超过峰值时覆盖峰值

3.3.4 列表点击事件将对应item加入库内人员,名字由编号对应生成,添加成功时如果设备定位开启,重新更新设备定位参数

3.3.5 搜索栏为空或者纯空格时点击搜索会恢复成所有数据

3.4 com.position4g.ui.fragment.Device\_HomeFragment

3.4.1 onSyncMessage 同步状态与定位小区来自两个不同的json

获取同步状态:DataUtils.getPaTemBean(event.message)

{“deviceNumber”:“12324”,"patemp":[{"paNumber":PA编号1,"temp":当前温度},{"paNumber":PA编号2,"temp":当前 温度}],"devtemp":设备温度,"boardtemp":主板温度,"wirelessSwitch":无线开关状态,"sms Switch":短息模块开关状态,"syncinfo":[{"cellNumber":BBU模块编号1,"syncmethod":当前 同步的方式,"syncstatus":同步状态},{"cellNumber":BBU模块编号2,"syncmethod":当前同 步的方式,"syncstatus":同步状态}]，"smsManagerSwitch":当前短信维护开关状态,"paPower":[{"paNumber":PA编号1,"power":当前功率},{"paNumber":PA编号2,"power":当前功 率}],"paWaveRatio":[{"paNumber":PA编号1,"waveRadio":当前驻波比},{"paNumber":PA编 号2,"waveRadio":当前驻波比}],"paSwitch":[{"paNumber":PA编号1,"paSwitch":当前PA开 关状态},{"paNumber":PA编号2,"paSwitch":当前PA开关状态}],"autoSniffer":自动扫频开 关状态,"cpu":CPU使用率,"memory":内存使用率,"uptime":上报时间戳}

获取定位小区 DataUtils.getBBUBean(event.message)

0不可用, 1可用 2可用并且定位小区

{"bbuStatus" : {"bbu2":1 , "bbu3":2}, "RadioState":{"Radio1": 1, "Radio2": 0}, "TddLocationEnable": 1, "FddLocationEnable": 0}

当接收到两种json使得同步状态与定位小区参数都不为空时才算作有效javabean

3.4.2 点击按钮开启定位时,会将库内人员列表数组发送

开启/关闭电信/联通共用一条json,通过切换联通/电信按钮区分

电信/联通按钮同时应只有一个处于可点击状态,由管理服务器发送过来的数据决定

开启定位：

{“code”: 4608, "DeplexMode":"FDD" ,"imsiList" :[{“imsi”: 13246798332 }, {"imsi": 124354271223}]，"timestamp":1470819628,"type":4192}

注：DeplexMode为FDD时如果设备是CM\_CU则开启联通定位，如果设备是CM\_CT则开启电信定位；如果DeplexMode为TDD则开启移动定位，imsiList最多可携带6个

关闭定位：

{“code”:4609,"DeplexMode":"FDD","timestamp":1470819628,"type":4192}

切换定位:

{"code":0x1207, "data" : {"deviceType" : 8, timestamp":1470819628,"type":4193}} (8和9分别对应联通和电信)

根据result判断是否更改成功

3.5 com.position4g.ui.fragment.DeviceParamFragment设备参数页面

提交参数修改接口:

{"code" : 4617, "data" : [{ "cellNumber":2 ,"freqList": "1"},{"cellNumber":3 ,"freqList": "38094,37900,38950", "gpsDelay":-70000, "isCMLocaiton": 0},{ "cellNumber":6 ,"freqList": "100", "CtFreqList": "1650", "gpsDelay":-70000, "isCULocaiton": 1, "isCTLocaiton": 1}], timestamp":1470819628,"type":4193}

(cellNumber 为6或者7的时候freqList 代表的是联通的频点列表，CtFreqList 为电信的频点列表，

isCMLocaiton为1时指定该移动基带板作为定位小区

isCULocaiton指定联通的定位小区

isCTLocaiton指定电信的定位小区

FDD和TDD各只能指定一块基带板作为定位小区 ，可以不指定定位小区)

最后一个点击修改的城市为激活状态,下次打开app时自动切换至改城市,信息存储在

PreferenceUtils.putString(Constants.ACTIVIE\_CITY, activeCity);

点击修改按钮后将会保存数据至数据库,不管是否为激活状态,下次切换至该城市后将会从数据库取值,否则使用默认模版.点击’取消’按钮将数据恢复为最后一次保存的该城市数据库数据,若未保存过则恢复模版

radiogroup放置了一个invisible的radiobutton,是为了按取消时回复未选中状态(移动频段默认未选中)

3.6 com.position4g.ui.fragment.DeviceSyncFragment同步模式与功率输出

数据交互待测试

配置基带板输出：

{"code" : 4611, "data":{"cellNumber":1, "value":25}

注:一次只能携带一个cellnumber,不能用数组的形式统一发送,所以每个bbu设置一个单独的按钮

设置基带板同步模式：

{"code" : 4617, "data":{"cellNumber":1, "syncMode":3}, "timestamp":1470819628,"type":4193} （0为空口同步适合没有gps信号时使用，1为gps同步模式，3为混合同步模式，默认混合同步）

这里没有存数据库或者sharedpreference,总是显示默认B1/混合

3.7 com.position4g.ui.fragment.MemberLibFragment 库内人员界面

com.position4g.ui.fragment.MemberLocalFragment 本地人员界面

3.7.1 Silencesms相关代码已标注文档注释

3.7.1.1 核心代码:

发送sms:SmsManager.getDefault().sendDataMessage(mTvPhone.getText().toString().trim(), null, (short) 9200, payload, sentPI, deliveryPI);

接受广播:

if (DELIVER.equalsIgnoreCase(intent.getAction())) {

//实际上移动号码走到这一步就基本是在线了,不在线一般不会收到这个intent

boolean delivered = false;

if (intent.hasExtra("pdu")) {

byte[] pdu = (byte[]) intent.getExtras().get("pdu");

if (pdu != null && pdu.length > 1) {

String resultPdu = getLogBytesHex(pdu).trim();

delivered = "00".equalsIgnoreCase(resultPdu.substring(resultPdu.length() - 2));

}

}

mTvState.setText(delivered ? "在线" : "不在线");

if(delivered){

tempDatas.get(checkPosition).setState(1);

adapter.notifyDataSetChanged();

}

}

3.7.2 重写点击事件,单击列表item弹出在线信息窗口,长按显示/隐藏下方按钮

3.7.3 库内加入本地/本地加入库内都做了去重处理,已存在的imsi不会覆盖或重复新增,如果要更新数据要先删除原有数据,或者重写逻辑

3.7.4 如果定位开启,删除/新增库内人员都会将首页设备页面数据重新发送并携带新的库内人员数组

3.7.5 dialog.setCancelable(false);这一句使得新增人员窗口无法点击外部窗口/cancel键而消失.如果不需要改为true

3.7.6新增人员时将备注保存至数据库了,但没有做展示,需要展示可以创建textview并调用MemberBean.getRemark()

3.7.7 本地人员界面同上

3.8 com.position4g.ui.fragment.PositionFragment 定位状态页面

功率(dbm)参数需要与com.position4g.ui.fragment.DeviceOutputFragment

页面配合测试一下

{“deviceNumber”:“12324”,"patemp":[{"paNumber":PA编号1,"temp":当前温度},{"paNumber":PA编号2,"temp":当前 温度}],"devtemp":设备温度,"boardtemp":主板温度,"wirelessSwitch":无线开关状态,"sms Switch":短息模块开关状态,"syncinfo":[{"cellNumber":BBU模块编号1,"syncmethod":当前 同步的方式,"syncstatus":同步状态},{"cellNumber":BBU模块编号2,"syncmethod":当前同 步的方式,"syncstatus":同步状态}]，"smsManagerSwitch":当前短信维护开关状态,"paPower":[{"paNumber":PA编号1,"power":当前功率},{"paNumber":PA编号2,"power":当前功 率}],"paWaveRatio":[{"paNumber":PA编号1,"waveRadio":当前驻波比},{"paNumber":PA编 号2,"waveRadio":当前驻波比}],"paSwitch":[{"paNumber":PA编号1,"paSwitch":当前PA开 关状态},{"paNumber":PA编号2,"paSwitch":当前PA开关状态}],"autoSniffer":自动扫频开 关状态,"cpu":CPU使用率,"memory":内存使用率,"uptime":上报时间戳}

3.9 com.position4g.ui.fragment.WarnFragment告警信息页面

定位设备新增告警信息：

{"alarmType": int , "data":{"alarmCode": int,"msg":""}," timestamp":1470819628} (alarmType代表1为set，2为clear；alarmCode代表alarm类型， msg代表alarm信息)

alarmcode:

1 bbu 重启

2 MAC\_PHY\_ABNORMAL

3 BBU 断线

接受到告警信息时调用((HomeActivity)getActivity()).setWarnTag(),左侧drawerlayout列表告警信息选项文字右侧出现小红点,点击后消失

新增告警字体颜色设置为红色,已解决告警字体设置为绿色

3.10 com.position4g.ui.fragment.ScanFragment扫描模式页面

待测试

打开扫描模式：

{"code":4618,"data":{"scanPeriod":50},"timestamp":1470819628,"type":4192}

单位为秒，不能少于15秒

关闭扫描模式：

{"code" : 4619, "timestamp":1470819628,"type":4192}

3.11 com.position4g.ui.fragment.Setting\_Fragment设置页面

3.11.1切换帧码记录完整时间会改变adapter的item布局(否则完整时间会挤出屏幕或者不完整时间左边缩进右边留白不好看),此布局改动对列表所有已存在item都生效,但已存在的item数据未做更改(遍历消耗资源过大),会导致之前的数据格式不好看

3.11.2 开启省电模式:如果bbu定位小区状态为否,开启定位时将此bbu关闭,默认是开启定位时打开所有bbu

3.12 com.position4g.ui.activity.LoginActivity 登录界面

3.12.1初始帐号密码都为admin

3.12.2 PreferenceUtils.getBoolean(Constants.DEFAULTPASSWORD,true)为true时,表示密码没有修改过为默认的admin,修改密码后取值为false

1. 自定义控件

4.1 com.position4g.ui.dialog.AddLibDialog 库内人员管理界面弹出的增加人员对话框

4.1.1 ButterKnife.bind需要手动调用,插件默认生成的位置会报空指针异常

4.1.2 builder.setCancelable(false)在这里设置无效,需要在父布局fragment中设置

4.2 com.position4g.ui.dialog.AddMemberDialog 本地人员管理界面弹出的增加人员对话框,同上

1. 全局上下文com.position4g.MyApp

全局唯一的application,静态方法getInstance()获取全局上下文context

1. Build.gradle注意:

6.1 TargetSDKVERSION 设置为22,不会触发运行时权限,只有silencesms处处理了运行时权限

6.2 com.github.PhilJay:MPAndroidChart 2.x版本和3.x版本API有大量改动,3.x版本网上教程资料匮乏,如需求变动建议改用2.x版本jar包