Framework层通用接口使用说明

# 一、Framework层中间接口设计

## 1．在Framework层添加系统服务octopu做为连接应用层和Framework层的接口。

com.android.server.octopu.OctopuSerivce

android.octopu.OctopuManager

## 2．应用层开发只关注使用OctopuManager接口,如更新传感器数据：

mOctopuManager = (OctopuManager) getSystemService("octopu");

Bundle bundle = new Bundle();

bundle.putInt(OctopuManager.SENSOR\_TYPE, 1);

bundle.putFloat(OctopuManager.SENSOR\_X, 1.01f);

bundle.putFloat(OctopuManager.SENSOR\_Y, 1.02f);

bundle.putFloat(OctopuManager.SENSOR\_Z, 1.03f);

bundle.putLong(OctopuManager.SENSOR\_TIME, SystemClock.elapsedRealtime());

mOctopuManager.upSensorData(bundle);

## 3．Framework可通过使用系统服务注册监听被动接收应用传来的数据，如实时接收更新的传感器数据：

mOctopuManager = (OctopuManager) getSystemService("octopu");

listener = new OctopuManager.senserListener() {

@Override

public void notifySensorChange(Bundle bundle) {

//TODO::

int type = bundle.getInt(OctopuManager.SENSOR\_TYPE);

float x = bundle.getFloat(OctopuManager.SENSOR\_X);

float y = bundle.getFloat(OctopuManager.SENSOR\_Y);

float z = bundle.getFloat(OctopuManager.SENSOR\_Z);

long time = bundle.getLong(OctopuManager.SENSOR\_TIME);

}

};

mOctopuManager.addSensorListener(listener);

mOctopuManager.removeSensorListener(listener);//NOTE:注意释放资源，否则会内存泄漏

# 二、OctopuManager接口说明

## 1．传感器。

更新传感器数据

**void upSensorData(Bundle bundle);**

获取最后设置的传感器数据

**Bundle getSensorData();**

实时获取数据添加以下接口的监听

public void addSensorListener(SenserListener senserListener) ;

public void removeSensorListener(SenserListener senserListener);

public interface SenserListener {

void notifySensorChange(Bundle bundle);

}

## 2．GPS

更新GPS数据

**void upGpsData(Bundle bundle);**

获取最后设置的GPS数据

**Bundle getGpsData();**

实时获取数据添加以下接口的监听

public void addGpsListener(gpsListener gpsListener) ;

public void removeGpsListener(gpsListener gpsListener);

public interface GpsListener {

void notifyGpsChange(Bundle bundle);

}

设置定位调用示例：

Bundle gpsData = new Bundle();  
Location loc = new Location("gps");  
loc.setLongitude(location[0]);//经度  
loc.setLatitude(location[1]);//纬度  
loc.setAccuracy(10.0F);//定位精度

loc.setAltitude(200.0F);//高度

loc.setBearing(20.0F);//方位  
loc.setSpeed(1.0F);//速度M/S  
loc.setTime(SystemClock.*elapsedRealtime*());  
loc.setElapsedRealtimeNanos(SystemClock.*elapsedRealtimeNanos*());  
gpsData.putParcelable(OctopuManager.*GPS\_LOCATION*,loc);  
mOctopuManager.upGpsData(gpsData);

## 3．基站信息

更新基站数据

**void upCellData(Bundle bundle);**

获取最后设置的基站数据

**Bundle getCellData();**

实时获取数据添加以下接口的监听

public void addCellListener(CellListener cellListener) ;

public void removeCellListener(CellListener cellListener);

public interface CellListener {

void notifyCellChange(Bundle bundle);

}

设置基站信息，清空基站列表，可以传入空ArrarList队列。

Bundle cellData = new Bundle();  
cellData.putParcelableArrayList(OctopuManager.*CELL\_LIST*,new ArrayList<>());  
mOctopuManager.upCellData(cellData);

## 4．WIFI信息

更新WIFI数据

**void upWifiData(Bundle bundle);**

获取最后设置的WIF数据

**Bundle getWifiData();**

实时获取数据添加以下接口的监听

public void addWifiListener(WifiListener wifiListener) ;

public void removeWifiListener(WifiListener wifiListener);

public interface WifiListener {

void notifyWifiChange(Bundle bundle);

}

## 5．电话本添加/删除

更新电话本数据

**void** upPhonebookData(Bundle bundle)

实时获取数据添加以下接口的监听

public addPhonebookListener(PhonebookListener phonebookListener);

public removePhonebookListener(PhonebookListener phonebookListener);

public interface PhonebookListener {

void notifyPhonebookChange(Bundle bundle);

}

添加电话本：

OctopuManager.upPhonebookData(bundle)

OctopuManager.PB\_ACTION :

OctopuManager.PB\_ACTION\_SAVE = 0: 保存，如果不设置默认为保存

OctopuManager.PB\_NAME: 名字

OctopuManager.PB\_NUMBER： 号码

OctopuManager.PB\_SIMCARD： true保存在sim卡中，false保存在手机中

例子：

OctopuManager octopuManager = (OctopuManager) getSystemService("octopu");

Bundle bundle = new Bundle();

bundle.putInt(OctopuManager.PB\_ACTION, OctopuManager.PB\_ACTION\_SAVE);

bundle.putString(OctopuManager.PB\_NAME, "名字");

bundle.putString(OctopuManager.PB\_NUMBER, "1234567890");

bundle.putBoolean(OctopuManager.PB\_SIMCARD, true);

octopuManager.upPhonebookData(bundle);

删除电话本：

1. OctopuManager.upPhonebookData(bundle)

OctopuManager.PB\_ACTION:

OctopuManager. PB\_ACTION\_DELETE\_ONE = 1: 删除单个

OctopuManager. PB\_ACTION\_DELETE\_ALL = 2: 删除全部

OctopuManager.PB\_NAME: 名字

OctopuManager.PB\_NUMBER： 号码

OctopuManager.PB\_SIMCARD： true删除在sim卡中，false删除在手机中

例子：

删除单个电话本：

OctopuManager octopuManager = (OctopuManager) getSystemService("octopu");

Bundle bundle = new Bundle();

bundle.putInt(OctopuManager.PB\_ACTION,

OctopuManager. PB\_ACTION\_DELETE\_ONE);

bundle.putString(OctopuManager.PB\_NAME, "名字");

bundle.putString(OctopuManager.PB\_NUMBER, "1234567890");

bundle.putBoolean(OctopuManager.PB\_SIMCARD, true);

octopuManager.upPhonebookData(bundle);

删除全部：

OctopuManager octopuManager = (OctopuManager) getSystemService("octopu");

Bundle bundle = new Bundle();

bundle.putInt(OctopuManager.PB\_ACTION,

OctopuManager. PB\_ACTION\_DELETE\_ALL);

bundle.putBoolean(OctopuManager.PB\_SIMCARD, true);

octopuManager.upPhonebookData(bundle);

## 6．飞行模式开关

void setAirplaneModeOn(boolean enabling)

true ： 开启飞行模式

false： 关闭飞行模式

例子：

OctopuManager octopuManager = (OctopuManager) getSystemService("octopu");

octopuManager. setAirplaneModeOn(true);