

云兔安卓 SDK 与蜂窝网络模块通信开发指导

云兔安卓 SDK 与蜂窝网络模块通信的开发主要下面三个方法

1. 通过 AT 命令获取，AT 命令主要依赖 Phone 对象，具体路径 android.internal.telephony.Phone.java。
2. 通过 android framework 中其他模块方直接获取，例如 cellid 和 lac 可以直接通过 telephony 管理模块获取等。
3. 通过 AT 命令获取，AT 命令直接与 TTY 设备进行交互（其他硬件平台备用）。

方法1、Phone对象下提供了下面两个接口供客户收发AT使用

```
void invokeOemRilRequestRaw(byte[] data, Message response);  
void invokeOemRilRequestStrings(String[] strings, Message response);
```

下面主要使用 invokeOemRilRequestStrings 方法来实现一个 AT COMMAND API 来供云兔业务使用

1、使用 invokeOemRilRequestStrings 实现 sendATcommand 接口，示例代码如下：

```
public static Handler mResponseHandler = new Handler() {  
    @Override  
    public void handleMessage(Message msg) {  
        for (Field f:fields){  
            Log.i(TAG,"fields : "+f.getName());  
        }*/  
        try {  
            Field exceptionField = msg.obj.getClass().getField("exception");  
            Exception exception= (Exception) exceptionField.get(msg.obj);  
            if (exception != null) {  
                Log.e(TAG, " RIL response emessage exception : "+exception);  
                return;  
            }  
            Field result = msg.obj.getClass().getField("result");  
            String[] results = (String[]) result.get(msg.obj);  
            switch (msg.what) {  
                //judge by cmdID  
                case 1:  
                    Response_ICCID =  
results[0].substring(results[0].indexOf("+ICCID:")+8);  
                    Log.i(TAG,"Response RIL ICCID:" + Response_ICCID);  
                    break;  
                case 2:  
                    ...  
  
                default:  
                    break;  
            }  
        } catch (Exception e) {  
            e.printStackTrace();  
            Log.e(TAG, " RIL response message exception : "+e.getMessage());  
        }  
    }  
};
```

```
private static void IoTLinksendATcommand(String cmd1,String cmd2,int cmdID) {
    if (mPhone == null) {
        Log.i(TAG, "mPhone null");
        return;
    }
    Method invokeOemRilRequestStrings = null;
    try {
        String command[] = {cmd1,cmd2};
        Class[] param = new Class[2];
        param[0] = String[].class;
        param[1] = Message.class;
        Class[] p = new Class[]{String[].class, Message.class};
        for (Method method : mPhone.getClass().getMethods()) {
            if (method.getName().equals("invokeOemRilRequestStrings")) {
                invokeOemRilRequestStrings = method;
            }
        }
        invokeOemRilRequestStrings.setAccessible(true);
        invokeOemRilRequestStrings.invoke(mPhone, command,
mResponseHandler.obtainMessage(cmdID));
    } catch (Exception e) {
        e.printStackTrace();
        Log.i(TAG, "send AT command exception : " + e.getMessage());
    }
}
```

2、使用扩展的 IoTLinksendATcommand 获取 SIM 卡的 ICCID 号

```
// 函数说明：获取当前卡的ICCID
// 入参：无
// 返回：当前卡的ICCID, string类型
public static String getICCID() {
    IoTLinksendATcommand("AT+ICCID", "+ICCID:", 1)
    Log.d(TAG, "Response_ICCID ");
    return Response_ICCID;
}
```

方法2、android framework 中其他模块方法直接获取

- 1、获取设备 IMEI 号, 使用 telephony 模块的 telephonyManager.getDeviceId 方法
- 2、获取 SIM 卡 IMSI 号, 使用 telephony 模块的 telephonyManager.getSubscriberId 方法
- 3、获取 lac cellid,使用 telephony 模块的 TelephonyManager.getSystemService 方法
- 4、获取 CSQ 信号强度, 使用 telephony 模块的 TelephonyManager.getSystemService 方法
- 5、获取 GPS 定位信息, 使用 locationManager 模块 (待详细确定方法)
- 6、获取运营商信息, 使用 telephony 模块, TelephonyManager.getSimOperatorName() 就可以直接获取到运营商的名字; 建议使用 TelephonyManager.getSimOperator() 获取 IMSI, 然后根据返回运营商
- 7、获取网络制式, 使用 telephony 模块的 telephonyManager.getNetworkType 方法

方法3、通过 AT 命令获取, AT 命令直接与 TTY 设备进行交互

获取 AT 通道的 tty 权限, 进行读写