**PRACTICAL # 13**

**OBJECT:**

Working with Networks and Telephony

**THEORY:**

Android exposes a comprehensive set of classes and functions to manage telephony and networks of the devices.

**Telephony**

- Initiating phone calls

- Reading the phone, network, data connectivity, and SIM states

- Monitoring changes to the phone, network, data connectivity.

**SMS**

- Send SMS and MMS messages

- Handling incoming SMS messages

android.telephony.TelephonyManager class provides information about the telephony services such as subscriber id, sim serial number, phone network type, the phone state etc.

The Android telephony APIs allows to

- Access the underlying telephone hardware stack

- Create your own dialer

- Integrate call handling and phone state monitoring

**Dialer**

The Dialer is an standard way for an app to make phone calls.

Intent.ACTION\_DIAL is used to launch dialer activity. Specify the number to dial using tel: schema as the data component of the Intent. It allows you to manage the call initialization (the default dialer asks the user to explicitly initiate the call). This functionality does not require any permissions.

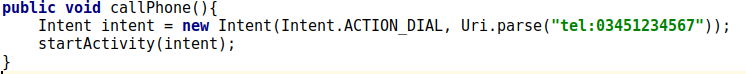


Figure 1: Dialing phone number.

**Reading phone, network, data connectivity, and SIM states**

Using TelephonyManager you can obtain:

- Phone type (GSM or CDMA),

- Unique device ID (IMEI or MEID),

- Software version,

- Phone number.

Access to telephony APIs is managed by the Telephony Manager. The code below gets handle to Telephony Manager.

*String srvcName = Context.TELEPHONY\_SERVICE;*

*TelephonyManager telephonyManager = (TelephonyManager)getSystemService(srvcName);*

This also requires the READ\_PHONE\_STATE permission in the manifest file.

*<uses-permission android:name="android.permission.READ\_PHONE\_STATE"/>*

Reading Phone Details

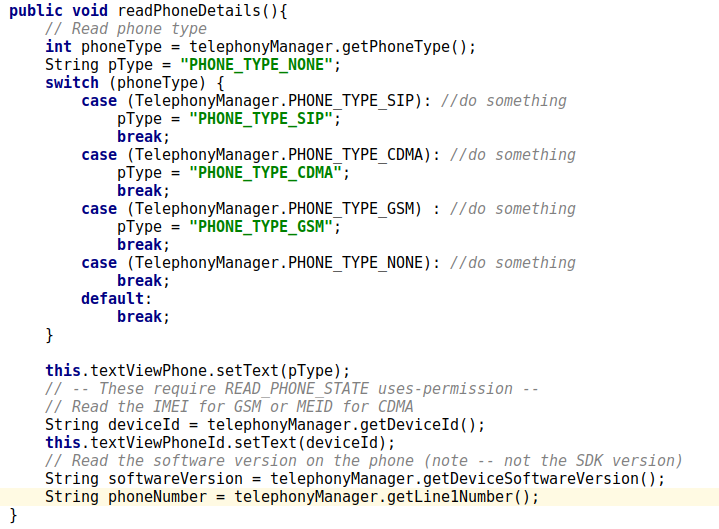


Figure 2: Reading phone details

The code in Figure 2 reads phone details including phone type, IMEI, software version and phone number.

**Network Details:**

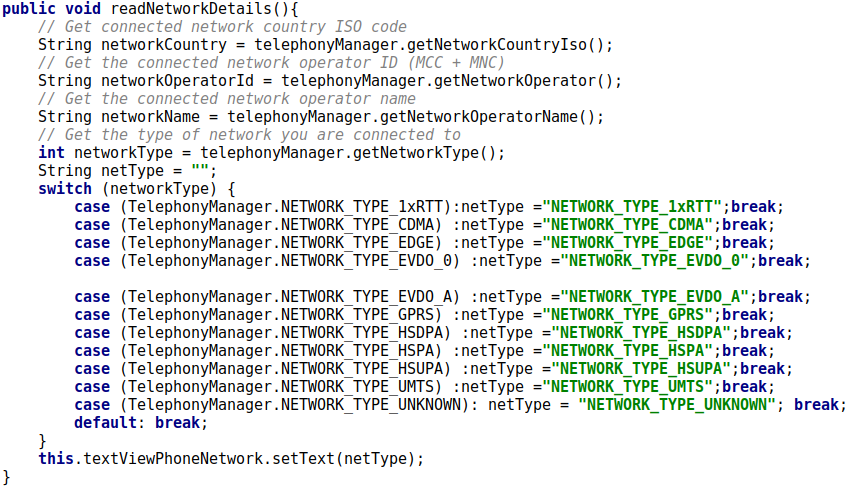


Figure 3: Reading network details.

In figure 3, we read network details including network country, operator id, network name, and network type. While SIM status details code are shown in Figure 4.

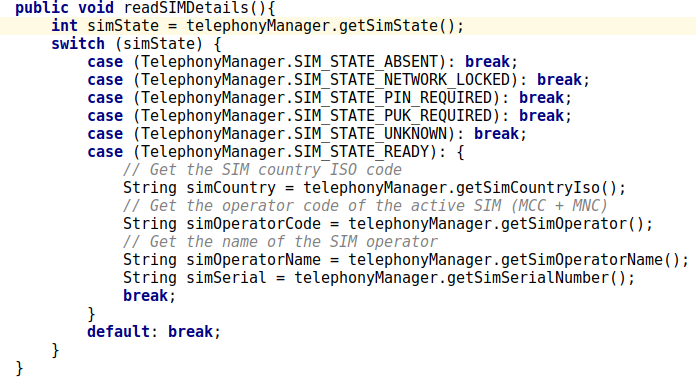


Figure 4: Reading SIM details

**Monitoring Phone Status:**

Extend PhoneStateListener class to listen and respond to; Phone state change events including call state (ringing, off hook, etc.), Cell location changes, Voice-mail and call-forwarding status, Phone service changes, Changes in mobile signal strength.

This requires READ\_PHONE\_STATE permission in Manifest file.

**Tracking Location Changes:**

onCellLocationChanged listens for cell location changes.

Handler receives a CellLocation object that includes methods for extracting the cell ID (getCid) and the current LAC (getLac).

Needs permission: <uses-permission android:name= "android.permission.ACCESS\_COARSE\_LOCATION"/>

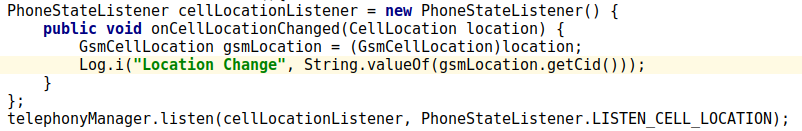


Figure 5: Listening to phone location change

**Sending SMS Using SmsManager:**

SMS messaging in Android is handled by the SmsManager.

*SmsManager smsManager = SmsManager.getDefault();*

Requires SEND\_SMS permission

Use sendTextMessage method of SmsManager, with the address (phone number) of the recipient and the text message you want to send (Figure 6).

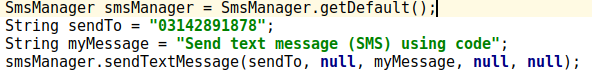


Figure 6: Sending SMS

**Handling Incoming SMS:**

For an incoming SMS, a broadcast Intent is fired with the android.provider.Telephony.SMS\_RECEIVED action. This requires RECEIVE\_SMS manifest permission.

Use the pdu extras key to extract an array of SMS PDUs each of which represents an SMS message SmsMessage.createFromPdu converts each PDU byte array into an SMS Message object. Figure 7 details handling of incoming SMS messages. We get a bundle object from the intent. Next we extract pdu objects from the bundle. Iterating the pdu objects, we get messages from the pdus. Finally sms text is extracted from the messages array.

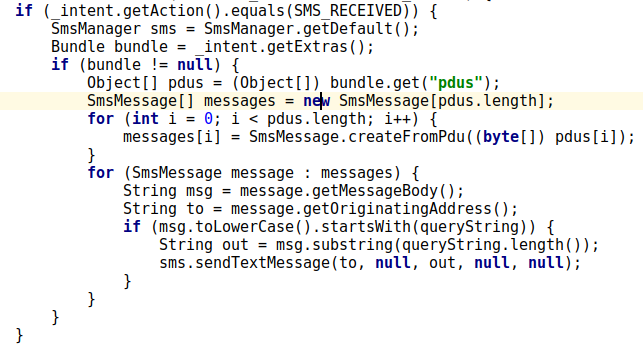


Figure 7: Handling incoming SMS

**ACTIVITIES**

**Activity 1**

Create an application that displays the device information, and network information.

**Activity 2**

Create an application allows user to make a phone call and send SMS.

**REVIEW QUESTIONS**

1. What is the main class used to manager telephony services of Android?
2. How can we make a phone call using telephony API?
3. How can monitor phone location changes in Android?
4. Why do we need to register an incoming SMS listener to catch incoming SMS?