



# SNS COLLEGE OF TECHNOLOGY

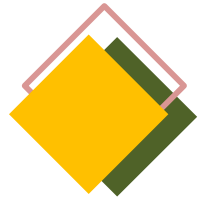
(An Autonomous Institution)

Re-accredited by NAAC with A+ grade, Accredited by NBA(CSE, IT, ECE, EEE & Mechanical)  
Approved by AICTE, New Delhi, Recognized by UGC, Affiliated to Anna University, Chennai



## Telephony & SMS API

Course: Mobile Application Development  
Unit : III – Building Blocks of Mobile Apps - II  
Class / Semester: II MCA / III Semester





- ❑ Android provides TelephonyManager API to implement telephony functionalities
- ❑ It include accessing network and device-type information, and retrieving information about phone state
- ❑ TelephonyManager is accessed through a telephony service provided by Android, by calling `getSystemService()` method

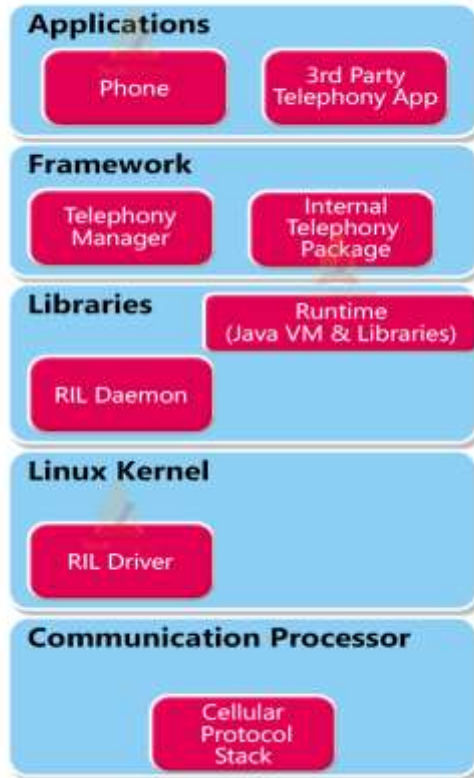
```
TelephonyManager telephonyManager =  
(TelephonyManager)getSystemService(Context.TELEPHONY_SERVICE);
```

- ❑ Access requires `READ_PHONE_STATE` permission to be included in manifest file
- ❑ Applications can also register a listener to receive notification of telephony state changes

Available in `android.telephony` package



## Android Telephony Framework



### ❑ Communication processor

- Collect and distribute data from peripherals (remote)
- Designed to communicate with the data communication network

### ❑ Radio Interface Layer(RIL) –

- interface through which hardware interact with framework. Two main components are
- **RIL Daemon**– It starts when the android system starts. It reads the system properties to find a library that is to be used for Vendor RIL
- **Vendor RIL**–Driver/library that is specific to each modem

### ❑ Framework services – contains packages and assists Telephony manager



- Initiating phone calls
- Reading phone, network, data connectivity and SIM states
- Monitoring changes to phone, network, data connectivity and SMS
- Using Intents to send SMS and MMS
- Using SMS Manager to send message
- Handling incoming message

Available in `android.telephony` package



- ❑ SmsManager APIs to implement SMS (Short Message Service) related functionalities like sending data, text and SMS message
- ❑ Two main classes SmsManager and SmsMessage
- ❑ Get SmsManager object by calling static method `SmsManager.getDefault()`  
`SmsManager.getDefault().sendTextMessage("phone number", srcaddress, "msg", PendingIntent sent, PendingIntent receive)`
- ❑ You can add permission by `android.permission.SEND_SMS` & `android.permission.RECEIVE_SMS` in manifest file

Available in `android.telephony` package



## activity\_main.xml

```
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:paddingBottom="@dimen/activity_vertical_margin"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    tools:context=".MainActivity" >
```

### <TextView

```
    android:id="@+id/textView1"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignParentLeft="true"
    android:layout_alignParentTop="true"
    android:layout_marginLeft="38dp"
    android:layout_marginTop="30dp"
    android:text="Phone Details:" />
```

### </RelativeLayout>



## MainActivity.java

```
public class MainActivity extends Activity {
    TextView textView1;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        textView1=(TextView)findViewById(R.id.textView1);
        TelephonyManager tm=(TelephonyManager)getSystemService(Context.TELEPHONY_SERVICE);
        String IMEINumber=tm.getDeviceId();
        String subscriberID=tm.getDeviceId();
        String SIMSerialNumber=tm.getSimSerialNumber();
        String networkCountryISO=tm.getNetworkCountryIso();
        String SIMCountryISO=tm.getSimCountryIso();
        String softwareVersion=tm.getDeviceSoftwareVersion();
        String voiceMailNumber=tm.getVoiceMailNumber();
```

```
String strphoneType="";
    int phoneType=tm.getPhoneType();
    switch (phoneType)
    {
        case (TelephonyManager.PHONE_TYPE_CDMA): strphoneType="CDMA";
            break;
        case (TelephonyManager.PHONE_TYPE_GSM): strphoneType="GSM";
            break;
        case (TelephonyManager.PHONE_TYPE_NONE): strphoneType="NONE";
            break;
    }
    boolean isRoaming=tm.isNetworkRoaming(
);
```



## MainActivity.java

```
String info="Phone Details:\n";
info+="\n IMEI Number:"+IMEINumber;
info+="\n SubscriberID:"+subscriberID;
info+="\n Sim Serial Number:"+SIMSerialNumber;
info+="\n Network Country ISO:"+networkCountryISO;
info+="\n SIM Country ISO:"+SIMCountryISO;
info+="\n Software Version:"+softwareVersion;
info+="\n Voice Mail Number:"+voiceMailNumber;
info+="\n Phone Network Type:"+strphoneType;
info+="\n In Roaming? :"+isRoaming;

textView1.setText(info);//displaying the information in t
he textView
}
```

## AndroidManifest.xml

provide **READ\_PHONE\_STATE** permission in the AndroidManifest.xml file.

```
permission android:name="android.permission.READ_PHONE_STATE"/>
```





## Example: Access Call state



```
TelephonyManager telephonyManager =  
    (TelephonyManager) getSystemService(Context.TELEPHONY_SERVICE);  
PhoneStateListener callStateListener = new PhoneStateListener() {  
public void onCallStateChanged(int state, String incomingNumber)  
{  
    if(state==TelephonyManager.CALL_STATE_RINGING){  
        Toast.makeText(getApplicationContext(),"Phone Is Ringing",  
            Toast.LENGTH_LONG).show();    }  
    if(state==TelephonyManager.CALL_STATE_OFFHOOK){  
        Toast.makeText(getApplicationContext(),"Phone is Currently in A call",  
            Toast.LENGTH_LONG).show(); }  
    if(state==TelephonyManager.CALL_STATE_IDLE){  
        Toast.makeText(getApplicationContext(),"phone is neither ringing nor in a call",  
            Toast.LENGTH_LONG).show(); }  
    }  
};  
telephonyManager.listen(callStateListener,PhoneStateListener.LISTEN_CALL_STATE);  
  
}
```



```
protected void onCreate(Bundle savedInstanceState) {  
    super.onCreate(savedInstanceState);  
    setContentView(R.layout.activity_main);  
    editText1=(EditText)findViewById(R.id.editText1);  
    button1=(Button)findViewById(R.id.button1);  
    button1.setOnClickListener(new OnClickListener(){  
        @Override  
        public void onClick(View arg0) {  
            String number=editText1.getText().toString();  
            Intent callIntent = new Intent(Intent.ACTION_CALL)  
;  
            callIntent.setData(Uri.parse("tel:"+number));  
            startActivity(callIntent);  
        }  
    });  
}
```

By setting permission in manifest file by  
<uses-  
permission android:name="android.per  
mission.CALL\_PHONE" />



Thank You

