



Android SMS

Introduction

- We can send SMS from our android application in two ways either by using **SMSManager** API or **Intents** based on our requirements.
- If we use **SMSManager** API, it will directly send SMS from our application. Using the SMS Manager, you can replace the native SMS application to send text messages, react to incoming texts, or use SMS as a data transport layer.
- In case if we use Intent with proper action (**ACTION_VIEW**), it will invoke a built-in SMS app to send SMS from our application.

Android Send SMS using SMSManager API

- To send SMS using SMSManager API we need to write the code like as shown below.

```
SmsManager smgr = SmsManager.getDefault();  
smgr.sendTextMessage(MobileNumber,null,Message,null,null);
```

- Following are the five arguments to the sendTextMessage() method:
 - destinationAddress—Phone number of the recipient
 - scAddress—Service center address; use null for default SMSC
 - text—Content of the SMS message
 - sentIntent—Pending intent to invoke when the message is sent
 - deliveryIntent—Pending intent to invoke when the message has been delivered
- **SMSManager** API required **SEND_SMS** permission in our android manifest to send SMS. Following is the code snippet to set **SEND_SMS** permissions in manifest file.

```
<uses-permission android:name="android.permission.SEND_SMS"/>
```

Sending SMS Messages Programmatically

- **Step 1:** Using Android Studio, create a new Android project and name it **SMS**.
- **Step 2 :** Replace the TextView with the following bolded statements in the activity_main.xml file.[Be sure to replace instances of com.jfdimarzio with the package used in your project:]

```
<?xml version="1.0" encoding="utf-8"?>
```

```
<LinearLayout xmlns:android=
```

```
http://schemas.android.com/apk/res/android
```

```
    android:layout_width="match_parent"
```

```
    android:layout_height="match_parent"
```

```
    android:orientation="vertical" >
```

```
    <Button
```

```
        android:text="Send SMS"
```

```
        android:layout_width="wrap_content"
```

```
        android:layout_height="wrap_content"
```

```
        android:id="@+id/btnSendSMS"
```

```
        android:onClick="onClick" />
```

```
</LinearLayout>
```

Sending SMS Messages Programmatically

- **Step 3:** In the AndroidManifest.xml file, add the following bolded statements.

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.jfdimarzio.sms">
    <uses-permission android:name="android.permission.SEND_SMS"/>
    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:supportsRtl="true"
        android:theme="@style/AppTheme">
        <activity android:name=".MainActivity">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
    </application>
```

Sending SMS Messages Programmatically

- **Step 4:** Add the following statements in bold to the MainActivity.java file:

```
import android.support.v4.app.ActivityCompat;  
import android.support.v4.content.ContextCompat;  
import android.support.v7.app.AppCompatActivity;  
import android.os.Bundle;  
import android.telephony.SmsManager;  
import android.view.View;
```

```
public class MainActivity extends AppCompatActivity {  
    final private int REQUEST_SEND_SMS = 123;  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_main);  
    }  
}
```

SMS

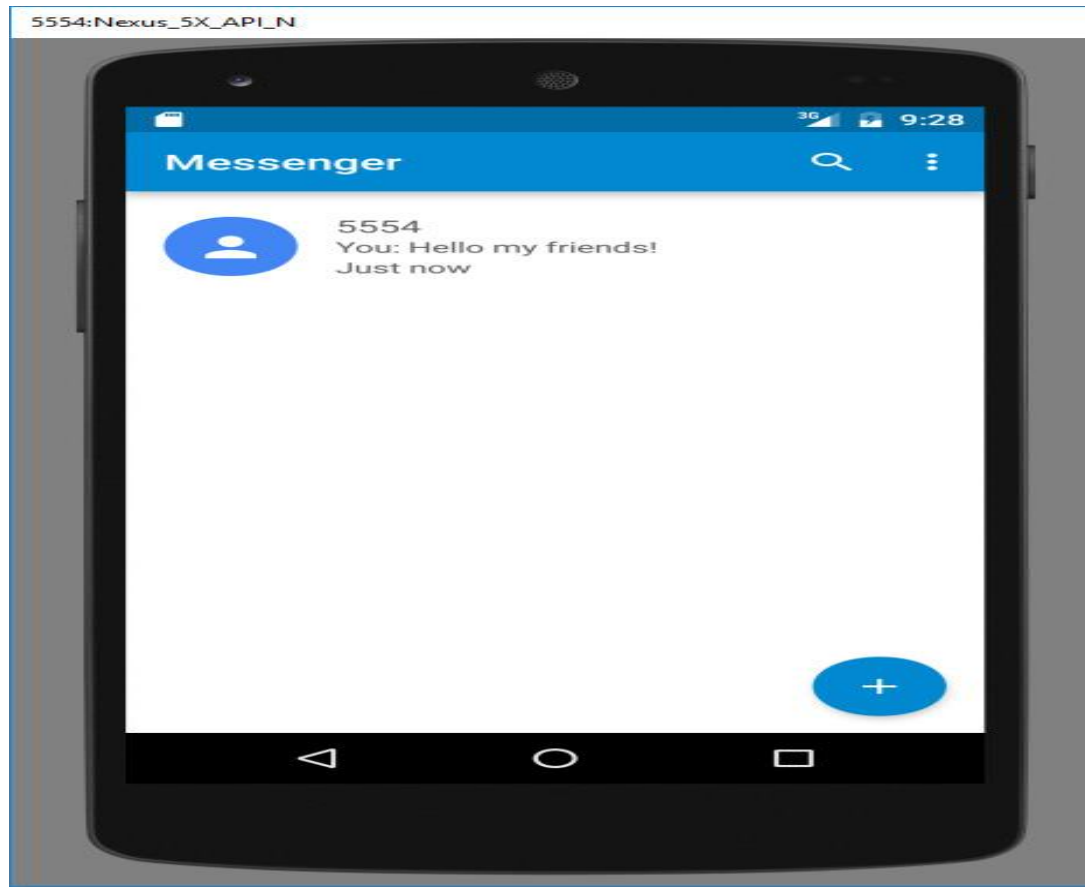
Sending SMS Messages Programmatically

```
public void onClick(View v) {  
    //---the "phone number" of your emulator should be 5554---  
    sendSMS("5554", "Hello my friends!");  
}  
  
//---sends an SMS message---  
private void sendSMS(String phoneNumber, String message)  
{  
    SmsManager sms = SmsManager.getDefault();  
    sms.sendTextMessage(phoneNumber, null, message, null, null);  
}  
}
```

SMS

Sending SMS Messages Programmatically

- **Step 5:** Press Shift+F9 to debug the application on the Android emulator.
- **Step 6 :** Click the Send SMS button to send an SMS message. Figure shows the SMS message received (view it by opening the messaging app on the emulator).



Receiving SMS Messages

- **Step 1:** Using the same project created in the previous example, add the following bolded statements to the AndroidManifest.xml file.

```
<?xml version="1.0" encoding="utf-8"?>
```

```
<manifest xmlns:android="http://schemas.android.com/apk/res/android"  
    package="com.jfdimarzio.sms">
```

```
    <uses-permission android:name="android.permission.SEND_SMS"/>
```

```
    <uses-permission android:name="android.permission.RECEIVE_SMS"/>
```

```
    <application
```

```
        android:allowBackup="true"
```

```
        android:icon="@mipmap/ic_launcher"
```

```
        android:label="@string/app_name"
```

```
        android:supportsRtl="true"
```

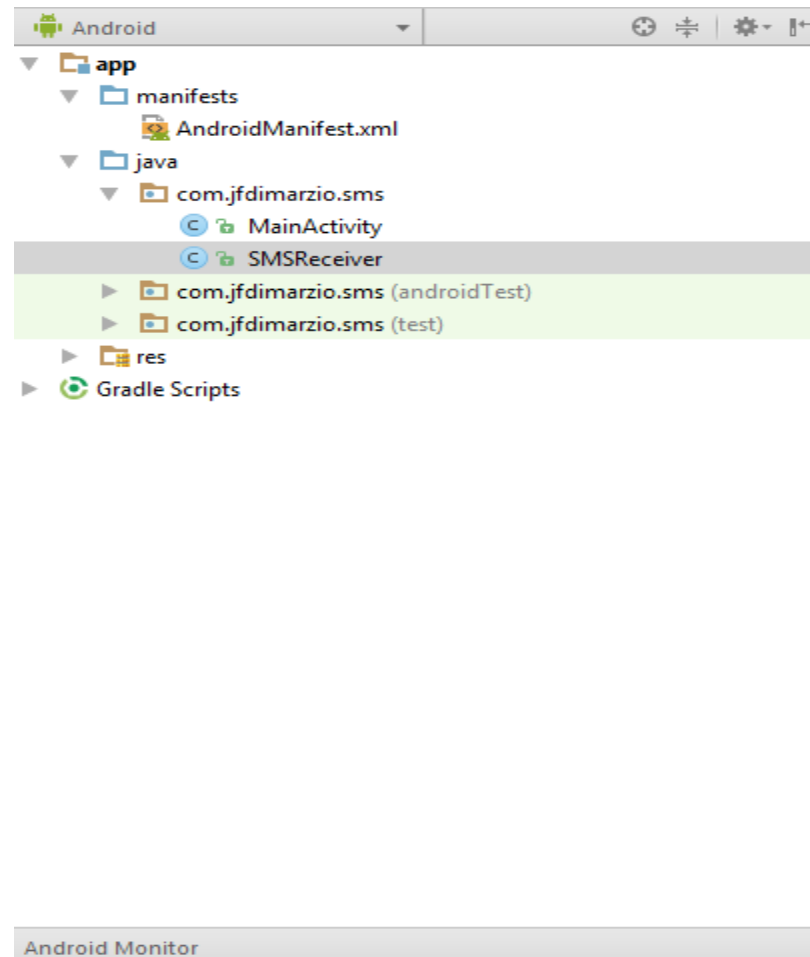
```
        android:theme="@style/AppTheme">
```

Receiving SMS Messages

```
<activity android:name=".MainActivity">
    <intent-filter>
        <action android:name="android.intent.action.MAIN" />
        <category android:name="android.intent.category.LAUNCHER" />
    </intent-filter>
</activity>
<receiver android:name=".SMSReceiver" android:exported="true"
    android:permission="android.permission.BROADCAST_SMS">
    <intent-filter android:priority="9000">
        <action
            android:name="android.provider.Telephony.SMS_RECEIVED" />
    </intent-filter>
</receiver>
</application>
</manifest>
```

Receiving SMS Messages

- **Step 2:** In the src folder of the project, add a new Class file to the package name and call it **SMSReceiver**



Receiving SMS Messages

- **Step 3:** Code the SMSReceiver.java file as follows:

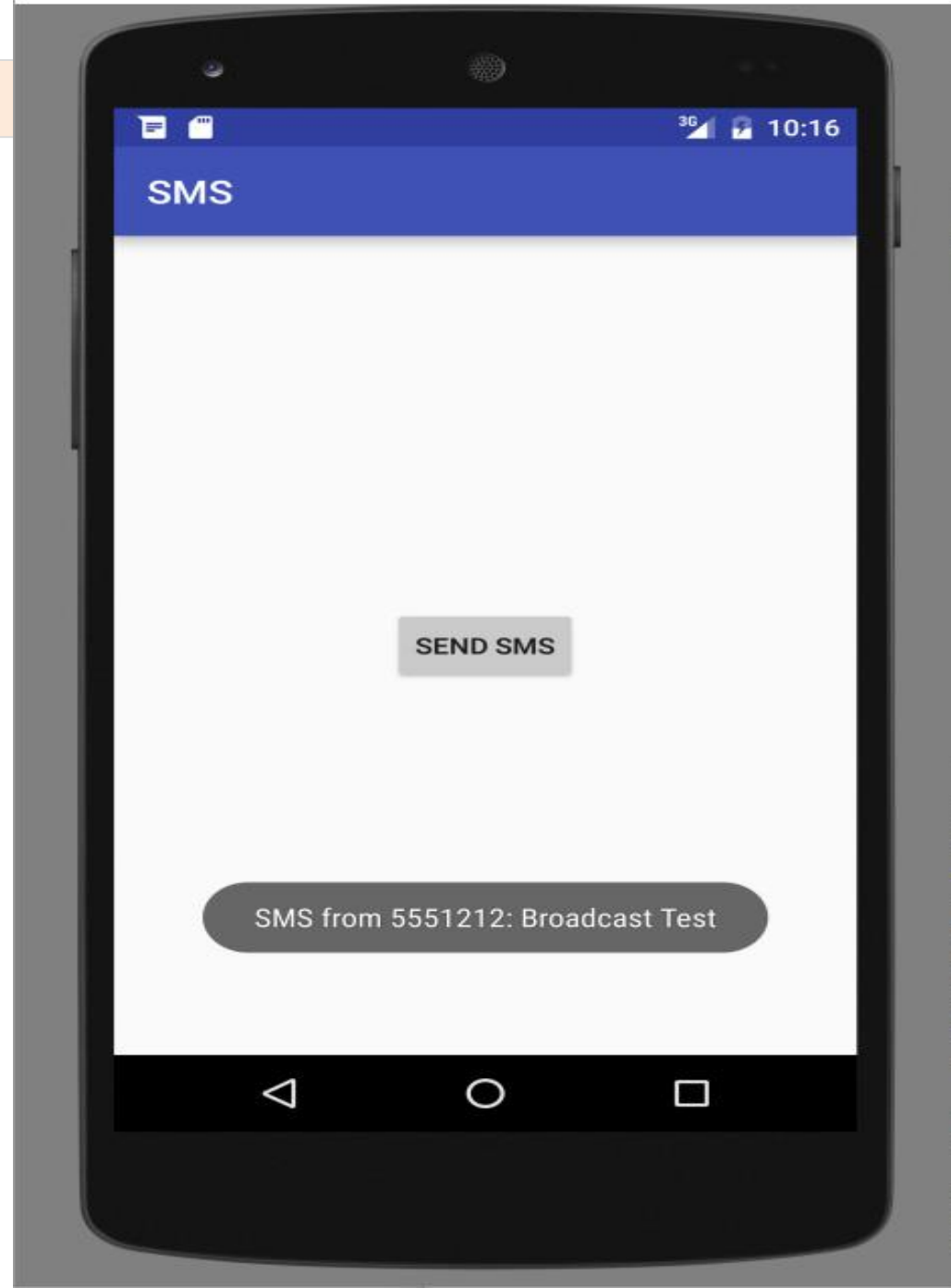
```
import android.content.BroadcastReceiver;
import android.content.Context;
import android.content.Intent;
import android.os.Bundle;
import android.telephony.SmsMessage;
import android.util.Log;
import android.widget.Toast;
public class SMSReceiver extends BroadcastReceiver
{
    @Override
    public void onReceive(Context context, Intent intent)
    {
        //---get the SMS message passed in---
        Bundle bundle = intent.getExtras();
        SmsMessage[] msgs = null;
        String str = "SMS from ";
        if (bundle != null)
```

Receiving SMS Messages

```
{
    //---retrieve the SMS message received---
    msgs = Telephony.Sms.Intents.getMessagesFromIntent(intent);
    for (int i=0; i<msgs.length; i++){
        str += msgs[i].getMessageBody().toString();
    }
    //---get the message body---
    str += msgs[i].getMessageBody().toString();
}
//---display the new SMS message---
Toast.makeText(context, str, Toast.LENGTH_SHORT).show();
Log.d("SMSReceiver", str);
}
}
}
```

Receiving SMS Messages

- **Step 4:** Press Shift+F9 to debug the application on the Android emulator.
- **Step 5:** Using the More setting on the emulator, select Phone and Send Message, send a message to the emulator. Your application should be able to receive the message and display it using the Toast class (see Figure) .



Android Send SMS using Intent

- Intent is a messaging object which is used to request an action from another app component such as activities, services, broadcast receivers, and content providers.
- To send SMS using the Intent object, we need to write the code like as shown below.

```
Intent sInt = new Intent(Intent.ACTION_VIEW);  
sInt.putExtra("address", new String[]{txtMobile.getText().toString()});  
sInt.putExtra("sms_body",txtMessage.getText().toString());  
sInt.setType("vnd.android-dir/mms-sms");
```

- Even for Intent, it required a **SEND_SMS** permission in our android manifest to send SMS. Following is the code snippet to set **SEND_SMS** permissions in manifest file.

```
<uses-permission android:name="android.permission.SEND_SMS"/>
```

Android Send SMS using Intent

- Call `startActivity` with an `Intent.ACTION_SENDTO` action Intent. Specify a target number using `sms: schema` notation as the Intent data. Include the message you want to send within the Intent payload using an `sms_body` extra:

```
Intent smsIntent = new Intent(Intent.ACTION_SENDTO,  
                             Uri.parse("sms:55512345"));  
smsIntent.putExtra("sms_body", "Press send to send me");  
startActivity(smsIntent);
```

- To attach files to your message (effectively creating an MMS message), add an `Intent.EXTRA_STREAM` with the URI of the resource to attach, and set the Intent type to the MIME type of the attached resource.
- Note that the native MMS application doesn't include an Intent Receiver for `ACTION_SENDTO` with a type set. Instead, you need to use `ACTION_SEND` and include the target phone number as an address extra:

Android Send SMS using Intent

```
// Get the URI of a piece of media to attach.  
Uri attached_Uri = Uri.parse("content://media/external/images/media/1");  
// Create a new MMS intent  
Intent mmsIntent = new Intent(Intent.ACTION_SEND, attached_Uri);  
mmsIntent.putExtra("sms_body", "Please see the attached image");  
mmsIntent.putExtra("address", "07912355432");  
mmsIntent.putExtra(Intent.EXTRA_STREAM, attached_Uri);  
mmsIntent.setType("image/jpeg");  
startActivity(mmsIntent);
```

Listening for Incoming SMS Messages

- When a device receives a new SMS message, a new Broadcast Intent is fired with the **android.provider.Telephony.SMS_RECEIVED** action.
- For an application to listen for SMS Broadcast Intents, it needs to specify the **RECEIVE_SMS** manifest permission:

<uses-permission android:name="android.permission.RECEIVE_SMS" />

- The SMS Broadcast Intent includes the incoming SMS details. To extract the array of `SmsMessage` objects packaged within the SMS Broadcast Intent bundle, use the `pdu` key to extract an array of SMS PDUs (protocol data units — used to encapsulate an SMS message and its metadata).
- To convert each PDU byte array into an SMS Message object, call `SmsMessage.createFromPdu`, passing in each byte array:

```
Bundle bundle = intent.getExtras();
if (bundle != null) {
    Object[] pdus = (Object[]) bundle.get("pdus");
    SmsMessage[] messages = new SmsMessage[pdus.length];
    for (int i = 0; i < pdus.length; i++)
        messages[i] = SmsMessage.createFromPdu((byte[]) pdus[i]);
}
```

Listening for Incoming SMS Messages

- To listen for incoming messages, register your SMS Broadcast Receiver using an Intent Filter that listens for the `android.provider.Telephony.SMS_RECEIVED` action String.
- Register this in the application manifest to ensure your application can always respond to incoming SMS messages.

```
<receiver android:name="MySMSReceiver">  
  <intent-filter>  
    <action android:name="android.provider.Telephony.SMS_RECEIVED"/>  
  </intent-filter>  
</receiver>
```

- Each `SmsMessage` contains the SMS message details, including the originating address (phone number), timestamp, and the message body, which can be extracted using the `getOriginatingAddress`, `getTimestampMillis`, and `getMessageBody` methods, respectively:

Listening for Incoming SMS Messages

```
public class MySMSReceiver extends BroadcastReceiver {  
    @Override  
    public void onReceive(Context context, Intent intent) {  
        Bundle bundle = intent.getExtras();  
        if (bundle != null) {  
            Object[] pdus = (Object[]) bundle.get("pdus");  
            SmsMessage[] messages = new SmsMessage[pdus.length];  
            for (int i = 0; i < pdus.length; i++)  
                messages[i] = SmsMessage.createFromPdu((byte[]) pdus[i]);  
            for (SmsMessage message : messages) {  
                String msg = message.getMessageBody();  
                long when = message.getTimestampMillis();  
                String from = message.getOriginatingAddress();  
                Toast.makeText(context, from + " : " + msg,  
                Toast.LENGTH_LONG).show();  
            }  
        }  
    }  
}
```

Android Send SMS Example

- The example to send SMS using **SMSManager** API in the android application.
- Create a new android application using android studio and give names as **SendSMSExample**. Now open activity_main.xml file and write the code below.

activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical" android:layout_width="match_parent"
    android:layout_height="match_parent">
    <TextView
        android:id="@+id/fstTxt"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginLeft="100dp"
        android:layout_marginTop="150dp"
        android:text="Mobile No" />
```

Android Send SMS Example

```
<EditText
    android:id="@+id/mbtTxt"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginLeft="100dp"
    android:ems="10"/>
<TextView
    android:id="@+id/secTxt"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Message"
    android:layout_marginLeft="100dp" />
<EditText
    android:id="@+id/msgTxt"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginLeft="100dp"
    android:ems="10" />
```

Android Send SMS Example

```
<Button  
    android:id="@+id/btnSend"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:layout_marginLeft="100dp"  
    android:text="Send SMS" />  
</LinearLayout>
```

Android Send SMS Example

- Now open our main activity file **MainActivity.java** and write the code like as shown below

MainActivity.java

```
package com.tutlane.sendsmsexample;
import android.content.Intent;
import android.net.Uri;
import android.provider.Telephony;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.telephony.SmsManager;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;
public class MainActivity extends AppCompatActivity {
    private EditText txtMobile;
    private EditText txtMessage;
    private Button btnSms;
```


Android Send SMS Example

@Override

```
protected void onCreate(Bundle savedInstanceState) {  
    super.onCreate(savedInstanceState);  
    setContentView(R.layout.activity_main);  
    txtMobile = (EditText)findViewById(R.id.mblTxt);  
    txtMessage = (EditText)findViewById(R.id.msgTxt);  
    btnSms = (Button)findViewById(R.id.btnSend);  
    btnSms.setOnClickListener(new View.OnClickListener() {  
        @Override  
        public void onClick(View v) {  
            try{  
                SmsManager smgr = SmsManager.getDefault();  
                smgr.sendTextMessage(txtMobile.getText().toString(),null,txtMes  
sage.getText().toString(),null,null);  
                Toast.makeText(MainActivity.this, "SMS Sent Successfully",  
Toast.LENGTH_SHORT).show();  
            }  
        }  
    });  
}
```

Android Send SMS Example

```
        catch (Exception e){
            Toast.makeText(MainActivity.this, "SMS Failed to Send, Please try
again", Toast.LENGTH_SHORT).show();
        }
    }
});
}
```

Android Send SMS Example

- Now open android manifest file (**AndroidManifest.xml**) and write the code like as shown below

AndroidManifest.xml

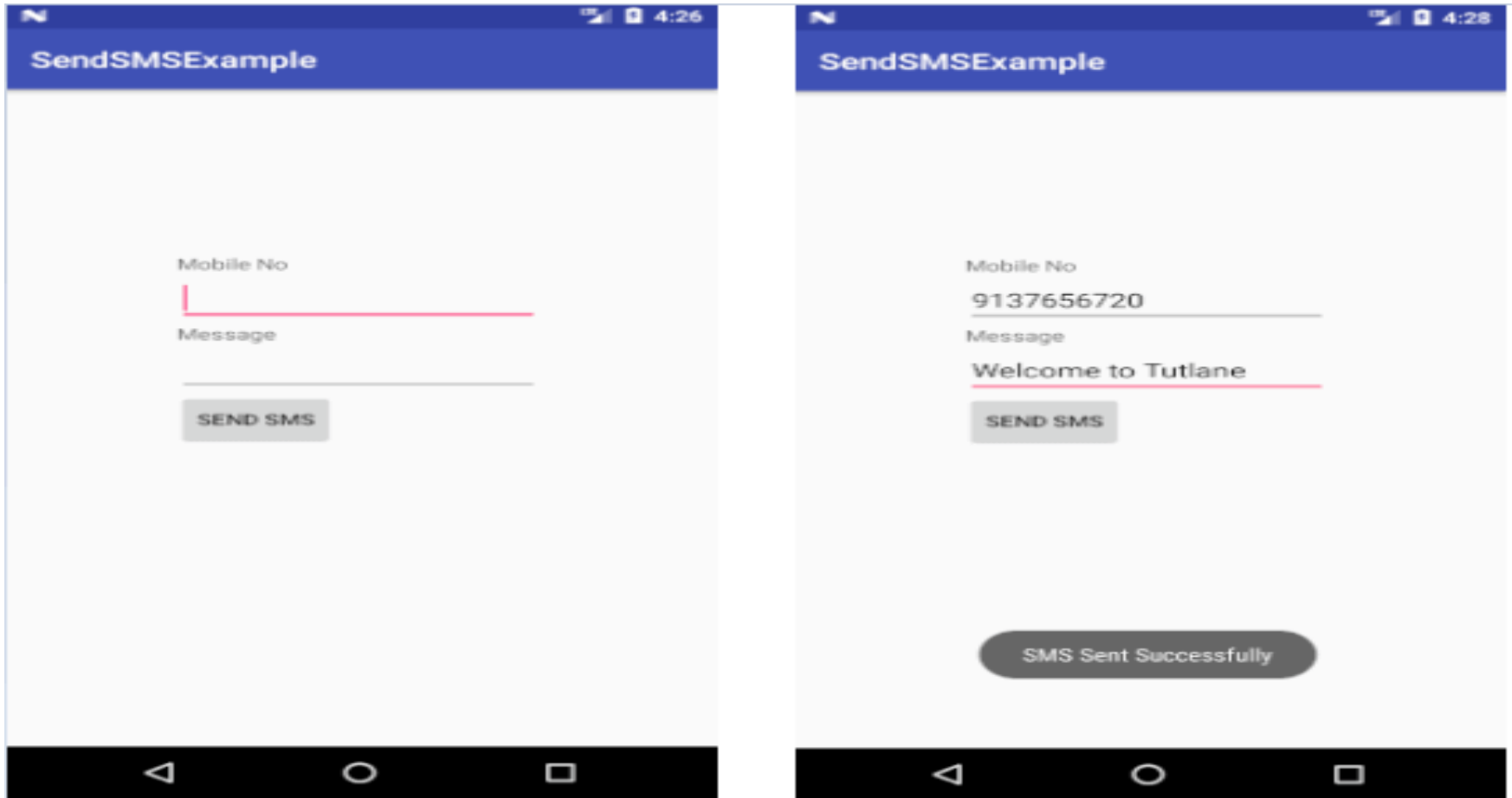
```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.tutlane.sendsmsexample">
    <uses-permission android:name="android.permission.SEND_SMS"/>
    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportsRtl="true"
        android:theme="@style/AppTheme">
        <activity android:name=".MainActivity">
```

Android Send SMS Example

```
<intent-filter>  
    <action android:name="android.intent.action.MAIN" />  
    <category android:name="android.intent.category.LAUNCHER" />  
</intent-filter>  
</activity>  
</application>  
</manifest>
```

SMS

Output of Android Send SMS Example



Once you enter all details and click on **Send SMS** button it will send SMS and show the alert message like as mentioned in above image.

Android Send SMS Example

- The above example we implemented using **SMSManager** API. In case if we want to use Intent to send SMS to replace button click code like as shown below.

```
btnSms.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        try{
            Intent i = new Intent(Intent.ACTION_VIEW);
            i.setData(Uri.parse("smsto:"));
            i.setType("vnd.android-dir/mms-sms");
            i.putExtra("address", new String(txtMobile.getText().toString()));
            i.putExtra("sms_body",txtMessage.getText().toString());
            startActivity(Intent.createChooser(i, "Send sms via:"));
        }
        catch(Exception e){
            Toast.makeText(MainActivity.this, "SMS Failed to Send, Please try again",
                Toast.LENGTH_SHORT).show();
        }
    }
});
```

Android Send SMS Example [using intent]

- Create a new android application using android studio and give names as **MyApplication**. Now open activity_main.xml file and write the code below.

activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?><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="16dp"    android:paddingLeft="16dp"
    android:paddingRight="16dp"    android:paddingTop="16dp"
    tools:context="com.example.hp.myapplication.MainActivity">
    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"    android:text="Send SMS"
        android:id="@+id/button"    android:onClick="sms"
        android:layout_alignParentBottom="true"
        android:layout_centerHorizontal="true"
        android:layout_marginBottom="114dp" />
</RelativeLayout>
```

Android Send SMS Example [using intent]

- Now open MainActivity.java file and write the code below.

MainActivity.java

```
import android.content.Intent;
import android.net.Uri;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.util.Log;
import android.view.View;
import android.widget.Toast;
public class MainActivity extends AppCompatActivity{
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
    }
}
```


Android Send SMS Example [using intent]

```

public void sms(View v) {
    Log.i("Sending SMS","");
    Intent I =new Intent(Intent.ACTION_VIEW);
    I.setData(Uri.parse("smsto:"));
    I.setType("vnd.android-dir/mms-sms");
    I.putExtra("address", new String ("1234567890"));
    I.putExtra("sms_body","Enter your Sms here..");
    try    {
        startActivity(I);
        finish();
        Log.i("Sms Send","");
    }
    catch(Exception e)    {
        Toast.makeText(MainActivity.this,"Sms not
        send",Toast.LENGTH_LONG).show();
    }
}

```

Android Send SMS Example [using intent]

- Now open **AndroidManifest.xml** file and write the code below.

AndroidManifest.xml

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.hp.myapplication">
    <uses-permission android:name="android.permission.CALL_PHONE"/>
    <uses-permission android:name="android.permission.ACCESS_WIFI_STATE"/>
    <uses-permission
        android:name="android.permission.CHANGE_WIFI_STATE"/>
    <uses-permission android:name="android.permission.SEND_SMS"/>
    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:supportsRtl="true"
        android:theme="@style/AppTheme">
```

Android Send SMS Example [using intent]

```
<activity android:name=".MainActivity">  
  <intent-filter>  
    <action android:name="android.intent.action.MAIN" />  
    <category android:name="android.intent.category.LAUNCHER" />  
  </intent-filter>  
</activity>  
</application>  
</manifest>
```

SMS

Output of Android Send SMS Example [using intent]

