**Lab 5: Android Send SMS**

# **Introduction**

Depending on our needs, we have two options for sending SMS from an android application: either using the SMSManager API or intents.If we use the SMSManager API, our application will be able to deliver SMSes directly. If we use an intent with the right action (ACTION\_VIEW), our application will be able to deliver SMS messages using the built-in SMS app.

**Let’s get Started:**

In this Exercise we will be implementing  to send SMS using **SMSManager** API in the android application..

**Step 1**: open an **activity\_main.xml** file from **\res\layout** path and write the code like as shown below.

**activity\_main.xml**

*<?*xml version="1.0" encoding="utf-8"*?>*<LinearLayout 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:orientation="vertical"  
 android:padding="4dp"  
 tools:context=".MainActivity">  
 <EditText  
 android:id="@+id/editTextNum"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:hint="Type Phone number"  
 android:textColor="#FFA000" />  
 <EditText  
 android:id="@+id/editTextMsg"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:hint="Type message"  
 android:textColor="#FFA000" />  
 <Button  
 android:id="@+id/btnSendMsg"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_gravity="center"  
 android:layout\_marginTop="10dp"  
 android:onClick="sendMessage"  
 android:text="Send SMS" />  
</LinearLayout>

**Step2:**

Open main [activity](https://www.tutlane.com/tutorial/android/android-activity-lifecycle) file MainActivity.kt and write the code like as shown below.

**MainActivity.kt**

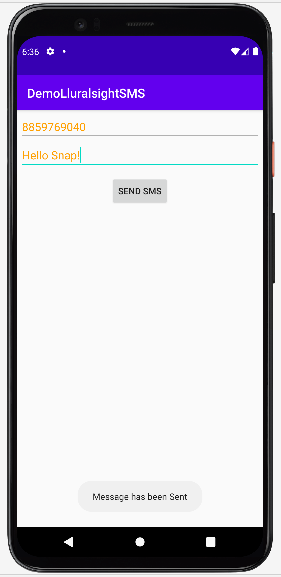
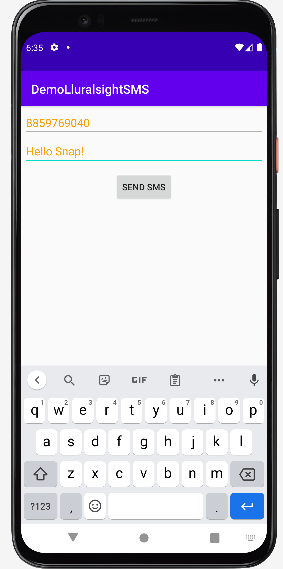
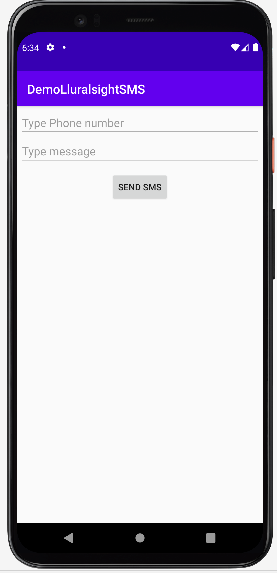
package com.example.demopluralsightsms  
import android.Manifest  
import android.content.pm.PackageManager  
import android.os.Bundle  
import android.support.v4.app.ActivityCompat  
import android.support.v4.content.ContextCompat  
import android.support.v7.app.AppCompatActivity  
import android.telephony.SmsManager  
import android.text.TextUtils  
import android.view.View  
import android.widget.Button  
import android.widget.EditText  
import android.widget.Toast  
class MainActivity : AppCompatActivity() {  
 lateinit var button: Button  
 lateinit var editTextNumber: EditText  
 lateinit var editTextMessage: EditText  
 private val permissionRequest = 101  
 override fun onCreate(savedInstanceState: Bundle?) {  
 super.onCreate(savedInstanceState)  
 setContentView(R.layout.*activity\_main*)  
 *title* = "DemoLluralsightSMS"  
 editTextNumber = findViewById(R.id.*editTextNum*)  
 editTextMessage = findViewById(R.id.*editTextMsg*)  
 button = findViewById(R.id.*btnSendMsg*)  
 }  
 fun sendMessage(view: View) {  
 val permissionCheck = ContextCompat.checkSelfPermission(this, Manifest.permission.*SEND\_SMS*)  
 if (permissionCheck == PackageManager.*PERMISSION\_GRANTED*) {  
 myMessage()  
 } else {  
 ActivityCompat.requestPermissions(this, *arrayOf*(Manifest.permission.*SEND\_SMS*),  
 permissionRequest)  
 }  
 }  
 private fun myMessage() {  
 val myNumber: String = editTextNumber.*text*.toString().*trim*()  
 val myMsg: String = editTextMessage.*text*.toString().*trim*()  
 if (myNumber == "" || myMsg == "") {  
 Toast.makeText(this, "Field should not be empty", Toast.*LENGTH\_SHORT*).show()  
 } else {  
 if (TextUtils.isDigitsOnly(myNumber)) {  
 val smsManager: SmsManager = SmsManager.getDefault()  
 smsManager.sendTextMessage(myNumber, null, myMsg, null, null)  
 Toast.makeText(this, "Message has been Sent", Toast.*LENGTH\_SHORT*).show()  
 } else {  
 Toast.makeText(this, "Please enter the correct number", Toast.*LENGTH\_SHORT*).show()  
 }  
 }  
 }  
 override fun onRequestPermissionsResult(requestCode: Int, permissions: Array<out String>, grantResults:  
 IntArray) {  
 super.onRequestPermissionsResult(requestCode, permissions, grantResults)  
 if (requestCode == permissionRequest) {  
 if (grantResults[0] == PackageManager.*PERMISSION\_GRANTED*) {  
 myMessage();  
 } else {  
 Toast.makeText(this, "Permission denied to send a message",  
 Toast.*LENGTH\_SHORT*).show();  
 }  
 }  
 }  
}

**Step 3:** 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"  
 xmlns:tools="http://schemas.android.com/tools">  
 <uses-permission android:name="android.permission.SEND\_SMS" />  
 <application  
 android:allowBackup="true"  
 android:dataExtractionRules="@xml/data\_extraction\_rules"  
 android:fullBackupContent="@xml/backup\_rules"  
 android:icon="@mipmap/ic\_launcher"  
 android:label="@string/app\_name"  
 android:roundIcon="@mipmap/ic\_launcher\_round"  
 android:supportsRtl="true"  
 android:theme="@style/Theme.DemoPluralSightSMS"  
 tools:targetApi="31">  
 <activity  
 android:name=".MainActivity"  
 android:exported="true">  
 <intent-filter>  
 <action android:name="android.intent.action.MAIN" />  
  
 <category android:name="android.intent.category.LAUNCHER" />  
 </intent-filter>  
  
 <meta-data  
 android:name="android.app.lib\_name"  
 android:value="" />  
 </activity>  
 </application>  
  
</manifest>

**Step 4: Check Output on Android Emulator and it should look like as given below.**

****

**Voila!!** We have successfully completed this lab.