**Lab 4: Android Phone Calls**

# **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 make a phone call by invoking the default phone calls app using an [Intent](https://www.tutlane.com/tutorial/android/android-intents-implicit-explicit) object 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"  
    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"  
        />  
    <EditText  
        android:id="@+id/mblTxt"  
        android:layout\_width="wrap\_content"  
        android:layout\_height="wrap\_content"  
        android:layout\_marginLeft="100dp"  
        android:ems="10">  
    </EditText>  
    <Button  
        android:id="@+id/btnCall"  
        android:layout\_width="wrap\_content"  
        android:layout\_height="wrap\_content"  
        android:layout\_marginLeft="100dp"  
        android:text="Call" />  
</LinearLayout>

**Step2:**

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

**MainActivity.java**

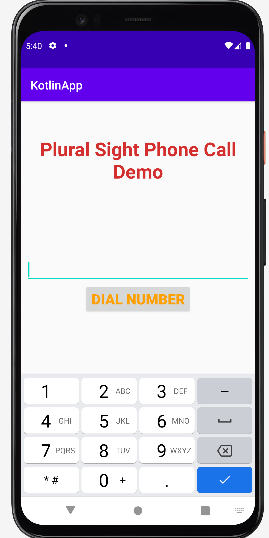
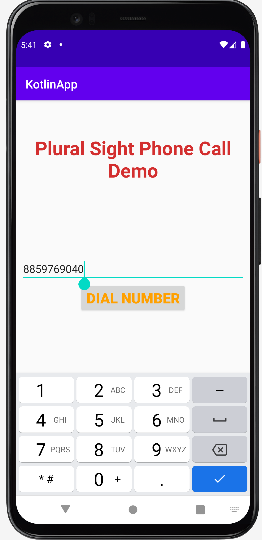
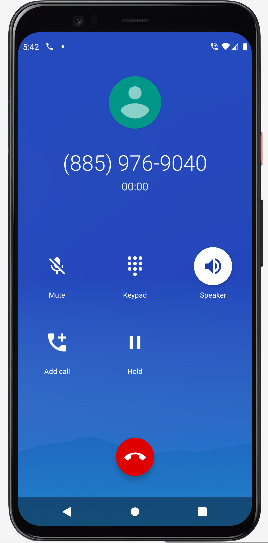
import android.Manifest;  
import android.content.Intent;  
import android.content.pm.PackageManager;  
import android.net.Uri;  
import android.os.Build;  
import android.support.v4.app.ActivityCompat;  
import android.support.v7.app.AppCompatActivity;  
import android.os.Bundle;  
import android.view.View;  
import android.widget.Button;  
import android.widget.EditText;  
  
public class MainActivity extends AppCompatActivity {  
    private EditText txtPhone;  
    private Button btn;  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity\_main);  
        txtPhone = (EditText)findViewById(R.id.mblTxt);  
        btn = (Button)findViewById(R.id.btnCall);  
  
        btn.setOnClickListener(new View.OnClickListener() {  
            @Override  
            public void onClick(View v) {  
                callPhoneNumber();  
            }  
        });  
    }  
    @Override  
    public void onRequestPermissionsResult(int requestCode, String[] permissions, int[] grantResults)  
    {  
        if(requestCode == 101)  
        {  
            if(grantResults[0] == PackageManager.PERMISSION\_GRANTED)  
            {  
                callPhoneNumber();  
            }  
        }  
    }  
    public void callPhoneNumber()  
    {  
        try  
        {  
            if(Build.VERSION.SDK\_INT > 22)  
            {  
                if (ActivityCompat.checkSelfPermission(this, Manifest.permission.CALL\_PHONE) != PackageManager.PERMISSION\_GRANTED) {  
                    ActivityCompat.requestPermissions(MainActivity.this, new String[]{Manifest.permission.CALL\_PHONE}, 101);  
                    return;  
                }  
                Intent callIntent = new Intent(Intent.ACTION\_CALL);  
                callIntent.setData(Uri.parse("tel:" + txtPhone.getText().toString()));  
                startActivity(callIntent);  
            }  
            else {  
                Intent callIntent = new Intent(Intent.ACTION\_CALL);  
                callIntent.setData(Uri.parse("tel:" + txtPhone.getText().toString()));  
                startActivity(callIntent);  
            }  
        }  
        catch (Exception ex)  
        {  
            ex.printStackTrace();  
        }  
    }  
}

**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"  
    package="com.tutlane.phonecallexample">  
    <uses-permission android:name="android.permission.CALL\_PHONE" />  
    <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">  
            <intent-filter>  
                <action android:name="android.intent.action.MAIN" />  
                <category android:name="android.intent.category.LAUNCHER" />  
            </intent-filter>  
        </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.