Sending SMS in Android and Receiving Feedback

Azim Zahir, 11 Jul 2013 CPOL

Rate this:

4.74 (21 votes)

This article explains how to send SMS in Android and receive sent and delivered feedback.

Download demo project - 43.5 KB

Introduction

Using Android programming, you can integrate SMS capabilities into your own Android applications. This allows you to create Android applications which can send and receive SMS messages from your Android device. In this article I am explaining how you can send SMS messages programmatically. Also I am showing how you can monitor the status of the SMS message, for example, when the message is sent and when it is delivered to the receiver.

Here, I am assuming that the reader has a basic knowledge of creating Android apps using the Eclipse IDE.

Background

Android requires permissions needed by an application to be specified in the *AndroidManifest.xml* file. This ensures that when the application is installed, the user knows which permissions are required by it. Also it gives an option to the user to decide whether or not to install the SMS application because such an application requires a user to incur the cost of sending SMS messages.

Using the Code

The following line is required in the *AndroidManifest.xml* file to allow the application to send SMS messages:

Hide Copy Code

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

The following is the full code of the *AndroidManifest.xml* file:

Hide Copy Code

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
      package="com.azim"
      android:versionCode="1"
      android:versionName="1.0">

    <uses-permission android:name="android.permission.SEND_SMS"/>
    <application android:icon="@drawable/icon" android:label="@string/app_name">
        <activity android:name=".SMSSenderActivity"
                  android:label="@string/app_name">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>

    </application>
</manifest>
```

The user interface of the SMS application consists of two `EditText` fields for accepting the message text and the phone number, respectively and a `Button` control to send the message. The following is the content of the *main.xml* file:

Hide Copy Code

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    >
<TextView
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:text="@string/hello"
    />
<TextView android:text="Enter SMS Text: " android:id="@+id/textView1"
  android:layout_width="wrap_content" android:layout_height="wrap_content"></TextView>
<EditText android:layout_width="match_parent" android:layout_height="wrap_content"
  android:id="@+id/editText1"></EditText>
<TextView android:text="Enter Phone Number: " android:id="@+id/textView2"
  android:layout_width="wrap_content" android:layout_height="wrap_content"></TextView>
<EditText android:layout_width="match_parent" android:layout_height="wrap_content"
  android:id="@+id/editText2"></EditText>
<Button android:text="Send SMS" android:id="@+id/button1" android:layout_width="wrap_content"
  android:layout_height="wrap_content"></Button>
</LinearLayout>
```

The `SmsManager` class is used to programmatically send an SMS message. This class is instantiated by using the `static getDefault()` method as follows:

Hide Copy Code

```
SmsManager sms=SmsManager.getDefault();
```

The `sendTextMessage()` method of the `SmsManager` class is used to send a text message as follows:

Hide Copy Code

```
sms.sendTextMessage(phone, null, message, piSent, piDelivered);
```

The `sendTextMessage()` method accepts five parameters, as follows:

- `phone` - Recipient's phone number
- `address` - Service Center Address (null for default)
- `message` - SMS message to be sent
- `piSent` - Pending intent to be invoked when the message is sent
- `piDelivered` - Pending intent to be invoked when the message is delivered to the recipient

The pending intents `piSent` and `piDelivered` are created as follows before calling the `sendTextMessage()` method:

Hide Copy Code

```
PendingIntent piSent=PendingIntent.getBroadcast(this, 0, new Intent("SMS_SENT"), 0);
PendingIntent piDelivered=PendingIntent.getBroadcast(this, 0, new Intent("SMS_DELIVERED"), 0);
```

The `PendingIntent` object `piSent` is used to notify the sender that the message has been sent and the `PendingIntent` object `piDelivered` is used to notify the sender that the message has been delivered to the recipient when the recipient actually receives the message.

Note: *The* `piDelivered PendingIntent` *does not fire in the Android emulator. You have to test the application on a real device to view it. However, the* `piSent PendingIntent` *works on both, the emulator as well as on a real device.*

Two `BroadcastReceiver` objects, `smsSentReceiver` and `smsDeliveredReceiver`, are created in the `onResume()` method. These are registered using the `registerReceiver()` method as follows:

Hide Copy Code

```
registerReceiver(smsSentReceiver, new IntentFilter("SMS_SENT"));
registerReceiver(smsDeliveredReceiver, new IntentFilter("SMS_DELIVERED"));
```

Inside each `BroadcastReceiver` object, the `onReceive()` method is overridden to check the result code using the `getResultCode()` method and display the appropriate message.

The two `BroadcastReceiver` objects are unregistered in the `onPause()` method as follows:

Hide Copy Code

```
unregisterReceiver(smsSentReceiver);
unregisterReceiver(smsDeliveredReceiver);
```

Following is the full source code of the application:

Hide Shrink Copy Code

```
package com.azim;

import android.app.Activity;
import android.app.PendingIntent;
import android.content.BroadcastReceiver;
import android.content.Context;
import android.content.Intent;
import android.content.IntentFilter;
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 SMSSenderActivity extends Activity implements View.OnClickListener {
    /** Called when the activity is first created. */
    EditText txtMessage,txtPhone;
    Button btnSend;
    BroadcastReceiver smsSentReceiver, smsDeliveredReceiver;
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);
        txtMessage=(EditText)findViewById(R.id.editText1);
        txtPhone=(EditText)findViewById(R.id.editText2);
        btnSend=(Button)findViewById(R.id.button1);
        btnSend.setOnClickListener(this);
    }

    public void onClick(View arg0) {
        // TODO Auto-generated method stub
        SmsManager sms=SmsManager.getDefault();
        String phone=txtPhone.getText().toString();
        String message=txtMessage.getText().toString();
        PendingIntent piSent=PendingIntent.getBroadcast(this, 0, new Intent("SMS_SENT"), 0);
        PendingIntent piDelivered=PendingIntent.getBroadcast(this, 0, new Intent("SMS_DELIVERED"), 0);
        sms.sendTextMessage(phone, null, message, piSent, piDelivered);
    }

    public void onResume() {
        super.onResume();
        smsSentReceiver=new BroadcastReceiver() {

            @Override
            public void onReceive(Context arg0, Intent arg1) {
                // TODO Auto-generated method stub
                switch (getResultCode()) {
                case Activity.RESULT_OK:
                    Toast.makeText(getBaseContext(), "SMS has been sent", Toast.LENGTH_SHORT).show();
                    break;
                case SmsManager.RESULT_ERROR_GENERIC_FAILURE:
                    Toast.makeText(getBaseContext(), "Generic Failure", Toast.LENGTH_SHORT).show();
```

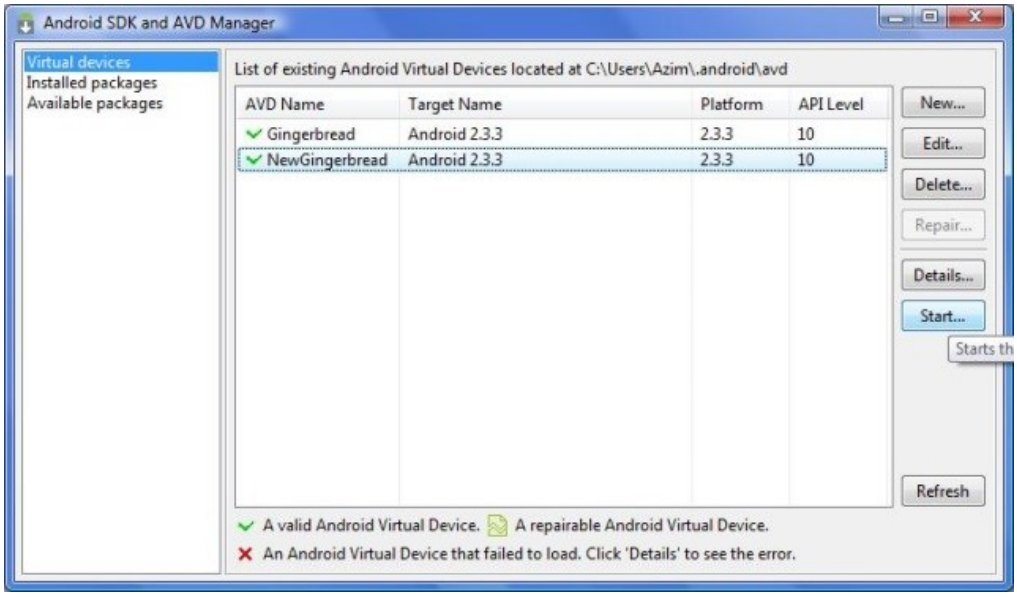
```
                    break;
                case SmsManager.RESULT_ERROR_NO_SERVICE:
                    Toast.makeText(getBaseContext(), "No Service", Toast.LENGTH_SHORT).show();
                    break;
                case SmsManager.RESULT_ERROR_NULL_PDU:
                    Toast.makeText(getBaseContext(), "Null PDU", Toast.LENGTH_SHORT).show();
                    break;
                case SmsManager.RESULT_ERROR_RADIO_OFF:
                    Toast.makeText(getBaseContext(), "Radio Off", Toast.LENGTH_SHORT).show();
                    break;
                default:
                    break;
                }

            }
        };
        smsDeliveredReceiver=new BroadcastReceiver() {

            @Override
            public void onReceive(Context arg0, Intent arg1) {
                // TODO Auto-generated method stub
                switch(getResultCode()) {
                case Activity.RESULT_OK:
                    Toast.makeText(getBaseContext(), "SMS Delivered", Toast.LENGTH_SHORT).show();
                    break;
                case Activity.RESULT_CANCELED:
                    Toast.makeText(getBaseContext(), "SMS not delivered", Toast.LENGTH_SHORT).show();
                    break;
                }
            }
        };
        registerReceiver(smsSentReceiver, new IntentFilter("SMS_SENT"));
        registerReceiver(smsDeliveredReceiver, new IntentFilter("SMS_DELIVERED"));
    }

    public void onPause() {
        super.onPause();
        unregisterReceiver(smsSentReceiver);
        unregisterReceiver(smsDeliveredReceiver);
    }
}
```

Execute the application on the Android emulator. This will start the default AVD. Using the Android SDK and AVD Manager option, launch another AVD as follows:



On the first emulator (5554), type the message and number of the second emulator (5556) and click on the Send SMS button. This will show the message on the second emulator and the sent notification on the first emulator.

Points of Interest

The advantage of sending SMS programmatically is that you can send dynamically generated messages through your applications. Also you don't need a real device to test this feature. You can test the application on the emulator before transferring it to the real device.

License

Share

About the Author

Azim Zahir

Instructor / Trainer NIIT, India

India

I am a trainer by profession. Currently I am working with NIIT (Mumbai, India) as a Senior Faculty. I enjoy programming as a hobby. My favorite technologies are Flash, Flex and Silverlight.

Of late I have developed keen interest in WPF and Windows Mobile programming.

Apart from computers, my favorite pastime is bicycling.

You may also be interested in...

Android SMS/MMS API: Sending SMS/MMS & Receiving SMS & Reading SMS History

SAPrefs - Netscape-like Preferences Dialog

SMS Gateway using Android Phone

Window Tabs (WndTabs) Add-In for DevStudio

How to Send and Receive SMS with the Orange SMS API

WPF: If Carlsberg did MVVM Frameworks: Part 3 of n

Comments and Discussions

You must Sign In to use this message board.

Search Comments Go

Profile popups Spacing Relaxed Layout Normal Per page 25 Update

First Prev Next

My vote of 2	**Member 11592632**	**8-Apr-15 22:24**
Short and Sweet.	**Jatin Kulkarni**	**4-Jan-15 1:10**
Re: Short and Sweet.	Azim Zahir	25-Apr-15 2:50
gr8 Tutorial	**anish.k abraham**	**14-Dec-14 0:41**
Thank you.	**WuRunZhe**	**19-Sep-14 21:19**
Re: Thank you.	Azim Zahir	20-Sep-14 16:47
thank you	**Solarin81**	**15-Jul-14 13:31**
Re: thank you	Azim Zahir	20-Sep-14 16:47
Can you write an article on how to receive SMS messages?	**Member 10890987**	**17-Jun-14 14:38**
Re: Can you write an article on how to receive SMS messages?	Azim Zahir	23-Jun-14 22:55
Move from activity to fragment and get error.	**mat_rapit**	**14-Feb-14 3:55**
Re: Move from activity to fragment and get error.	Azim Zahir	23-Jun-14 22:53
"Enter Phone Number" to default	**mat_rapit**	**13-Feb-14 16:52**
Re: "Enter Phone Number" to default	Azim Zahir	23-Jun-14 23:11
Good tut...	**Aravinth A**	**31-Dec-13 2:00**
Re: Good tut...	Azim Zahir	3-Jan-14 17:11
nor working	**Member 10425400**	**25-Nov-13 22:35**
Re: nor working	Azim Zahir	27-Nov-13 1:25
no receiver	**azmat ahmed**	**8-Oct-13 10:40**
Good One!! Vote 5	**boss prabu**	**20-Aug-13 18:18**
Re: Good One!! Vote 5	Azim Zahir	27-Aug-13 6:18
My vote of 5	**David H. Wright**	**19-Aug-13 4:13**
Re: My vote of 5	Azim Zahir	19-Aug-13 17:42
Problem with intent	**Member 3334506**	**9-Aug-13 12:31**
Re: Problem with intent	Azim Zahir	10-Aug-13 7:47

Last Visit: 31-Dec-99 19:00 Last Update: 10-Feb-16 21:04 Refresh 1

General News Suggestion Question Bug Answer Joke Praise Rant Admin

Use Ctrl+Left/Right to switch messages, Ctrl+Up/Down to switch threads, Ctrl+Shift+Left/Right to switch pages.