stackoverflow

Questions   Tags   Users   Badges   Unanswered   Ask Question

Sending and Receiving SMS and MMS in Android (pre Kit Kat Android 4.4)

CAREERS 2.0
by stackoverflow

+ Have projects on Codeplex?
Import them easily to your profile

asked   **1 year ago**
viewed   **43192 times**
active   **19 days ago**

Linked

69   Send SMS in android

24   Android - SMS Broadcast receiver

35   How to send image via MMS in Android?

17   How to send the SMS more than 160 character?

11   Android SMS receiver not working

14   Detecting SMS incoming and outgoing

8   Sending long sms messages

8   Unable to send sms using SMSManager in Android

2   Android send SMS automatically on button click

6   Detecting MMS messages on Android

see more linked questions…

Related

I have figured out how to send and receive **SMS** messages. To send **SMS** messages I had to call the `sendTextMessage()` and `sendMultipartTextMessage()` methods of the `SmsManager` class. To receive **SMS** messages, I had to register a receiver in the `AndroidMainfest.xml` file. Then I had to override the `onReceive()` method of the `BroadcastReceiver`. I have included examples below.

MainActivity.java

```java
public class MainActivity extends Activity {
    private static String SENT = "SMS_SENT";
    private static String DELIVERED = "SMS_DELIVERED";
    private static int MAX_SMS_MESSAGE_LENGTH = 160;

    // ---sends an SMS message to another device---
    public static void sendSMS(String phoneNumber, String message) {

        PendingIntent piSent = PendingIntent.getBroadcast(mContext, 0, new Intent(SENT), 0);
        PendingIntent piDelivered = PendingIntent.getBroadcast(mContext, 0,new Intent(DELIVERED), 0);
        SmsManager smsManager = SmsManager.getDefault();

        int length = message.length();
        if(length > MAX_SMS_MESSAGE_LENGTH) {
            ArrayList<String> messagelist = smsManager.divideMessage(message);
            smsManager.sendMultipartTextMessage(phoneNumber, null, messagelist, null, null);
        }
        else
            smsManager.sendTextMessage(phoneNumber, null, message, piSent, piDelivered);
    }

    //More methods of MainActivity ...
}
```

SMSReceiver.java

```java
public class SMSReceiver extends BroadcastReceiver {
    private final String DEBUG_TAG = getClass().getSimpleName().toString();
    private static final String ACTION_SMS_RECEIVED = "android.provider.Telephony.SMS_RECEIVE
    private Context mContext;
    private Intent mIntent;

    // Retrieve SMS
    public void onReceive(Context context, Intent intent) {
        mContext = context;
        mIntent = intent;

        String action = intent.getAction();

        if(action.equals(ACTION_SMS_RECEIVED)){

            String address, str = "";
            int contactId = -1;

            SmsMessage[] msgs = getMessagesFromIntent(mIntent);
            if (msgs != null) {
                for (int i = 0; i < msgs.length; i++) {
                    address = msgs[i].getOriginatingAddress();
                    contactId = ContactsUtils.getContactId(mContext, address, "address");
                    str += msgs[i].getMessageBody().toString();
                    str += "\n";
                }
            }

            if(contactId != -1){
                showNotification(contactId, str);
            }

            // ---send a broadcast intent to update the SMS received in the
```

AndroidManifest.xml

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

    <uses-sdk
        android:minSdkVersion="16"
        android:targetSdkVersion="17" />

    <uses-permission android:name="android.permission.READ_CONTACTS" />
    <uses-permission android:name="android.permission.READ_PHONE_STATE" />
    <uses-permission android:name="android.permission.SEND_SMS" />
    <uses-permission android:name="android.permission.RECEIVE_SMS" />
    <uses-permission android:name="android.permission.READ_SMS" />
    <uses-permission android:name="android.permission.WRITE_SMS" />
    <uses-permission android:name="android.permission.RECEIVE_MMS" />
    <uses-permission android:name="android.permission.WRITE" />
    <uses-permission android:name="android.permission.VIBRATE" />
    <uses-permission android:name="android.permission.INTERNET" />
    <uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />

    <application
        android:debuggable="true"
        android:icon="@drawable/ic_launcher_icon"
        android:label="@string/app_name" >

        <activity
            //Main activity...
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
```

However, I was wondering if you could send and receive **MMS** messages in a similar fashion. After doing some research, many examples provided on blogs simply pass an Intent to the native Messaging application. I am trying to send an **MMS** without leaving my application. There doesn't seem to be a standard way of sending and receiving **MMS**. Has anyone gotten this to work?

Also, I am aware that the SMS/MMS ContentProvider is not a part of the official Android SDK , but I was thinking someone may have been able to implement this. Any help is greatly appreciated.

Update

I have added a BroadcastReceiver to the AndroidManifest.xml file to receive **MMS** messages

```xml
<receiver android:name="com.sendit.receivers.MMSReceiver" >
    <intent-filter>
        <action android:name="android.provider.Telephony.WAP_PUSH_RECEIVED" />

        <data android:mimeType="application/vnd.wap.mms-message" />
    </intent-filter>
</receiver>
```

In the MMSReceiver class, the onReceive() method is only able to grab the phoneNumber that the message was sent from. How do you grab other important things from an **MMS** such as the file path to the media attachment (image/audio/video), or the text in the **MMS**?

MMSReceiver.java

```java
public class MMSReceiver extends BroadcastReceiver {
    private final String DEBUG_TAG = getClass().getSimpleName().toString();
    private static final String ACTION_MMS_RECEIVED = "android.provider.Telephony.WAP_PUSH_RI
    private static final String MMS_DATA_TYPE = "application/vnd.wap.mms-message";

    // Retrieve MMS
    public void onReceive(Context context, Intent intent) {

        String action = intent.getAction();
        String type = intent.getType();

        if(action.equals(ACTION_MMS_RECEIVED) && type.equals(MMS_DATA_TYPE)){

            Bundle bundle = intent.getExtras();

            Log.d(DEBUG_TAG, "bundle " + bundle);
            SmsMessage[] msgs = null;
            String str = "";
            int contactId = -1;
            String address;

            if (bundle != null) {

                byte[] buffer = bundle.getByteArray("data");
                Log.d(DEBUG_TAG, "buffer " + buffer);
                String incomingNumber = new String(buffer);
                int indx = incomingNumber.indexOf("/TYPE");
                if(indx>0 && (indx-15)>0){
                    int newIndx = indx - 15;
                    incomingNumber = incomingNumber.substring(newIndx, indx);
                    indx = incomingNumber.indexOf("+");
                    if(indx>0){
                        incomingNumber = incomingNumber.substring(indx);
```

According to the Documentation of android.provider.Telephony:

Broadcast Action: A new text based SMS message has been received by the device. The intent will have the following extra values:

pdus - An Object[] od byte[]s containing the PDUs that make up the message.

The extra values can be extracted using getMessagesFromIntent(android.content.Intent). If a BroadcastReceiver encounters an error while processing this intent it should set the result code appropriately.

```java
@SdkConstant(SdkConstantType.BROADCAST_INTENT_ACTION)
public static final String SMS_RECEIVED_ACTION = "android.provider.Telephony.SMS_RECEIVED";
```

Broadcast Action: A new data based SMS message has been received by the device. The intent will have the following extra values:

pdus - An Object[] of byte[]s containing the PDUs that make up the message.

The extra values can be extracted using getMessagesFromIntent(android.content.Intent). If a BroadcastReceiver encounters an error while processing this intent it should set the result code appropriately.

```java
@SdkConstant(SdkConstantType.BROADCAST_INTENT_ACTION)
public static final String DATA_SMS_RECEIVED_ACTION = "android.intent.action.DATA_SMS_RECEIVE
```

Broadcast Action: A new WAP PUSH message has been received by the device. The intent will have the following extra values:

transactionId (Integer) - The WAP transaction ID

pduType (Integer) - The WAP PDU type

header (byte[]) - The header of the message

data (byte[]) - The data payload of the message

contentTypeParameters (HashMap<String,String>) - Any parameters associated with the content type (decoded from the WSP Content-Type header)

If a BroadcastReceiver encounters an error while processing this intent it should set the result code appropriately. The contentTypeParameters extra value is map of content parameters keyed by their names. If any unassigned well-known parameters are encountered, the key of the map will be 'unassigned/0x...', where '...' is the hex value of the unassigned parameter. If a parameter has No-Value the value in the map will be null.

```java
@SdkConstant(SdkConstantType.BROADCAST_INTENT_ACTION)
public static final String WAP_PUSH_RECEIVED_ACTION = "android.provider.Telephony.WAP_PUSH_F
```

Update #2

I have figured out how to pass extras in a `PendingIntent` to be received by a `BroadcastReceiver` :
Android PendingIntent extras, not received by BroadcastReceiver

However, the extra gets passed to the *SendBroadcastReceiver* not the *SMSReceiver*. How can I pass an extra to the *SMSReceiver*?

Update #3

Receiving MMS

So after doing more research I saw some suggestions of registering a `ContentObserver` . That way you can detect when there are any changes to the `content://mms-sms/conversations` Content Provider, consequently allowing you to detect incoming MMS. Here is the closest example to get this to work that I have found: Receiving MMS

However, there is a variable `mainActivity` of type `ServiceController` . Where is the `ServiceController` class implemented? Are there any other implementations of a registered `ContentObserver` ?

Sending MMS

As for sending MMS, I have come across this example: Send MMS

The problem is that I tried running this code on my Nexus 4, which is on Android v4.2.2, and I am receiving this error:

```
java.lang.SecurityException: No permission to write APN settings: Neither user 10099 nor current proces
```

The error gets thrown after querying the `Carriers` ContentProvider in the `getMMSApns()` method of the `APNHelper` class.

```
final Cursor apnCursor = this.context.getContentResolver().query(Uri.withAppendedPath(Carriers.CONT
```

Apparently you can't Read APNs in Android 4.2

What is the alternative for all those applications which use mobile data to perform operations (like sending MMS) and don't know the default APN setting present in the device?

Update #4

Sending MMS

I have tried following this example: Send MMS

As @Sam suggested in his answer:

 You have to add jsoup to the build path, the jar to the build path and import com.droidprism.*; To do that in android, add the jars to the libs directory first, then configure the project build path to use the jars already in the libs directory, then on the build path config click order and export and check the boxes of the jars and move jsoup and droidprism jar to the top of the build order.

So now I no longer get the SecurityException errors. I am testing now on a Nexus 5 on Android KitKat. After running the sample code it gives me a 200 response code after the call to

MMResponse mmResponse = sender.send(out, isProxySet, MMSProxy, MMSPort);

Have you taken a look at this tutorial before? maximbogatov.wordpress.com/2011/08/13/mms-in-android – HaemEternal Feb 8 '13 at 8:32

However, I checked with the person I tried sending the MMS to. And they said they never received the MMS.Yes I have. I tried piecing Maxim's answer together but am not able to get it to work. There are many classes in there that import android.provider.telephony which seems to be deprecated. – toobsco42 Feb 8 '13 at 8:40

android   sms   broadcastreceiver   android-manifest   mms

And presumably, after reading @Sahil's answer, you've also tried this:
stackoverflow.com/questions/2972845/... – HaemEternal Feb 8 '13 at 8:44

share | improve this question   edited Feb 16 at 16:29   community wiki
                                                                               16 revs
I'm not sure how to piece that answer together, although it looks very similar to @Sahil's answer. – toobsco42
toobsco42 Feb 24 '13 at 21:52

Hi @toobsco42 are you able to find the solution for all those query you mention above.. ? –
kamal_tech_view Apr 26 '13 at 12:37 ✎

show **4** more comments

---

5 Answers   active   oldest   **votes**

11

When I have the exact same problem you describe above (Galaxy Nexus on t-mobile USA) it is because mobile data is turned off.

In Jelly Bean it is: Settings > Data Usage > mobile data

+50

Note that I have to have mobile data turned on PRIOR to sending an MMS OR receiving one. If I receive an MMS with mobile data turned off, I will get the notification of a new message and I will receive the message with a download button. But if I do not have mobile data on prior, the incoming MMS attachment will not be received. Even if I turn it on after the message was received.

For some reason when your phone provider enables you with the ability to send and receive MMS you must have the Mobile Data enabled, even if you are using Wifi, if the Mobile Data is enabled you will be able to receive and send MMS, even if Wifi is showing as your internet on your device.

It is a real pain, as if you do not have it on, the message can hang a lot, even when turning on Mobile Data, and might require a reboot of the device.

And GO Through this code :

http://downloads.ziddu.com/downloadfiles/14286605/SendMMS3.zip

Hope this helps.

share | improve this answer

edited Feb 14 '13 at 13:30

answered Feb 14 '13 at 13:23

Manan Merevik Sharma
318 ●3 ●19

Also you must know that sending a SMS and MMS are 2 entirely different things in background. MMS is more of an internet based network service as it requires sending additional items(Media) with text. The given code works fine on a few devices that i've tested on. ps : you can ignore the NOKIA part. – Manan Merevik Sharma Feb 14 '13 at 13:28

When i run this example, in LogCat it prints: 02-24 13:32:40.872: V/SendMMSActivity(5686): TYPE_MOBILE_MMS not connected, bail 02-24 13:32:40.882: V/SendMMSActivity(5686): type is not TYPE_MOBILE_MMS, bail It also says: java.lang.SecurityException: No permission to write APN settings: Neither user 10099 nor current process has android.permission.WRITE_APN_SETTINGS. It looks like it can't execute this query: final Cursor apnCursor = this.context.getContentResolver().query(Uri.withAppendedPath(Carriers.CONTENT_UR I, "current"), null, null, null, null); Im testing on a Nexus 4. – toobsco42 Feb 24 '13 at 21:40

Also this is the same example that @Sahil provided. – toobsco42 Feb 24 '13 at 21:47

add comment

6

There is not official api support which means that it is not documented for the public and the libraries may change at any time. I realize you don't want to leave the application but here's how you do it with an intent for anyone else wondering.

```java
public void sendData(int num){
    String fileString = "..."; //put the location of the file here
    Intent mmsIntent = new Intent(Intent.ACTION_SEND);
    mmsIntent.putExtra("sms_body", "text");
    mmsIntent.putExtra("address", num);
    mmsIntent.putExtra(Intent.EXTRA_STREAM, Uri.fromFile(new File(fileString)));
    msIntent.setType("image/jpeg");
    startActivity(Intent.createChooser(mmsIntent, "Send"));

}
```

I haven't completely figured out how to do things like track the delivery of the message but this should get it sent.

You can be alerted to the receipt of mms the same way as sms. The intent filter on the receiver should look like this.

```xml
<intent-filter>
    <action android:name="android.provider.Telephony.WAP_PUSH_RECEIVED" />
    <data android:mimeType="application/vnd.wap.mms-message" />
</intent-filter>
```

share | improve this answer

edited Jan 23 '13 at 19:15

answered Jan 23 '13 at 18:17

user1959417
106 ●7

Doesn't this just launch the native Messaging app? – toobsco42 Jan 23 '13 at 18:53

1 yeah sorry about that. i just realized that you already knew how to do that. i did add how to receive mms though. – user1959417 Jan 23 '13 at 19:16

Thanks, I have recently been implementing part of the MMS BroadcastReceiver , and have used the Intent Filter that you have posted. I will update this question soon. – toobsco42 Jan 23 '13 at 19:41 ✎

add comment

2

I dont think there is any sdk support for sending mms in android. Look here Atleast I havent found yet. But a guy claimed to have it. Have a look at this post.

Send MMS from My application in android

share | improve this answer

answered Jan 22 '13 at 6:46

Sahil Mahajan Mj
5,741 ● 2 ● 12 ● 52

I looked at the comments of the androidbridge.blogspot.com post of the Nokia implementation, and it looks like many people are having issues getting this to work on their devices. – toobsco42 Jan 22 '13 at 7:13

@toobsco42 So, there might be no support for it yet. – Sahil Mahajan Mj Jan 22 '13 at 7:20

add comment

2

To send an mms for Android 4.0 api 14 or higher without permission to write apn settings, you can use this library: Retrieve mnc and mcc codes from android, then call

```
Carrier c = Carrier.getCarrier(mcc, mnc);
if (c != null) {
    APN a = c.getAPN();
    if (a != null) {
        String mmsc = a.mmsc;
        String mmsproxy = a.proxy; //"" if none
        int mmsport = a.port; //0 if none
    }
}
```

To use this, add Jsoup and droid prism jar to the build path, and import com.droidprism.*;

share | improve this answer

edited Oct 24 '13 at 11:36

community wiki
5 revs, 3 users 66%
Sam Adams

Hey @Sam, I added the .jar file to my project but am getting this error at the line that instantiates the Carrier object : java.lang.NoClassDefFoundError: com.droidprism.Carrier is that happening to you? – toobsco42 Jul 26 '13 at 20:54

no. You have to add jsoup to the build path, the jar to the build path and import com.droidprism.*; I'll edit the answer. To do that in android, add the jars to the libs directory first, then configure the project build path to use the jars already in the libs directory, then on the build path config click order and export and check the boxes of the jars and move jsoup and droidprism jar to the top of the build order. – Sam Adams Jul 27 '13 at 17:44 ✎

Adding the Jsoup .jar resolved the NoClassDefFoundError. I can now get the APN settings. The next step is to figure out how to send MMS. – toobsco42 Jul 30 '13 at 17:05

add comment

0

I dont understand the frustrations. Why not just make a broadcastreceiver that filters for this intent:

android.provider.Telephony.MMS_RECEIVED

I checked a little further and you might need system level access to get this (rooted phone).

share | improve this answer

edited Feb 27 at 22:15

community wiki
3 revs
j2emanue

Hey @j2emanue, The problem is after you receive this intent, how do you actually get the content of the MMS? If an MMS contains and image and text, how do you extract these components. – toobsco42 Feb 27 at 22:05

but i notice theres a byte array extra you can get if you do it it the way .i mentioned .... byte[] data = intent.getByteArrayExtra("data"); im not sure how to parse it though sorry. – j2emanue Feb 27 at 22:15

i am able to parse it. but all i can get is the subject, who the mms came from, and the contentlocation where the content of the mms is stored. however this url is not accessible. – toobsco42 Feb 27 at 22:17 ✎

add comment

Your Answer

B *I* | 🌐 66 {} 🖼 | ≔ ≔ ≣ ≣ | ↶ ↷

Not the answer you're looking for? Browse other questions tagged 🤖 android sms broadcastreceiver android-manifest mms **or ask your own question.**

🟠 question feed