Mobile Applications for Business

Master SIA/SDBIS

Octavian Dospinescu

2021

General topics

- GSM capabilities in Android
- Receive/intercept an SMS message in a programmatic way
- Class **BroadcastReceiver** – general anatomy
- **Intent** – the concept
- Specific rights/permissions in AndroidManifest.xml
- Future directions

The framework - premises

- interception of the messages received by our phone from another device;
- the initiative does not belong to our user anymore, but to a third party who uses his own mobile device;
- we have no control over that certain transactions but we rather should "be alert " to the eventual receipt of an SMS.

Specific terms to know

- **Intent** = an event that occurs in the application's life;
- a **BroadcastReceiver** is able to intercept events of type **Intent** which appear from the external environment or from the operating system;
- we can associate it (BroadcastReceiver) with the notion of trigger.

BroadcastReceiver – general anatomy

- A **BroadcastReceiver** implements the abstract method **onReceive()** in order to process the **Intents** that arrive.
- The arguments of the method are a **Context** and an **Intent.**
- The method returns **void**, but it can be used the method **setResult** to send back a specific result.

BroadcastReceiver – general use

- In practice, the implementation of a BroadcastReceiver is made by **defining a subclass** of the base class BroadcastReceiver, where we **define/implement the method onReceive() according to the needs of the application.**

BroadcastReceiver – specifications

Unlike the Activity class (which **has a graphical appearance** as a form displayed on the mobile device), the **BroadcastReceiver** class *has not a visible graphical representation* for the user.

It works in the "background" of the application, **being started when a specific Intent** appears.

BroadcastReceiver – a model of implementation

```
public class ReceptorSMS extends BroadcastReceiver {
        @Override
        public void onReceive(Context context, Intent intent) {
                // TODO Auto-generated method stub
                Log.i("din receiver","s-a declansat metoda onReceive");
                if(intent.getAction().equals("android.provider.Telephony.SMS_RECEIVED"))
                {
                                Log.i("din receiver", "Am primit un SMS!!!");
                                //aici urmeaza sa completam cu cod de prelucrare

                }
        }
}
```

BroadcastReceiver – important specifications!!!

- So that our application can **intercept** SMS messages, it is necessary **to specify this permission** in the file AndroidManifest.xml by using **a clause of type <uses-permissions>** similar to that which gave us the right to send SMS messages.

A complete list of base permissions in Android can be found here:
http://developer.android.com/reference/android/Manifest.permission.html

BroadcastReceiver – permissions

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
      package="smsTavy.com"
      android:versionCode="1"
      android:versionName="1.0">
    <uses-sdk android:minSdkVersion="8" />
    <uses-permission android:name="android.permission.SEND_SMS"></uses-permission>
    <uses-permission android:name="android.permission.RECEIVE_SMS"></uses-permission>
    <application android:icon="@drawable/icon" android:label="@string/app_name">
        <activity android:name=".AppSMStavyActivity"
                  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>
```

BroadcastReceiver – important specifications

- For our **BroadcastReceiver** (ReceptorSMS) to work, **it must be also declared in AndroidManifest.xml file.**

BroadcastReceiver – declaration in AndroidManifest

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
      package="smsTavy.com"
      android:versionCode="1"
      android:versionName="1.0">
    <uses-sdk android:minSdkVersion="8" />
    <uses-permission android:name="android.permission.SEND_SMS"></uses-permission>
    <uses-permission android:name="android.permission.RECEIVE_SMS"></uses-permission>
    <application android:icon="@drawable/icon" android:label="@string/app_name">
        <activity android:name=".AppSMStavyActivity"
                  android:label="@string/app_name">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
        <receiver android:name="ReceptorSMS">
                  …
                  …
        </receiver>
    </application>
</manifest>
```

BroadcastReceiver – important specifications!!!

- for the receiver that we just declared in AndroidManifest.xml file, it should also be specified *the list of Intent types that it "listens to“*;

- a **BroadcastReceiver** can intercept several types of "messages" of intent type, such as *an incoming phone call, receiving an SMS, receiving of an email, discharging the battery below a certain limit etc.*

Intents and filters (IntentFilter)

In Android it is necessary to **make a mapping between the BroadcastReceiver and a list (filter) of Intents.**

This mapping is achieved through the concept of **IntentFilter**, which must also be declared in AndroidManifest.xml.

Intents and filters (IntentFilter)

Through the mapping between the BroadcastReceiver and the Intent, we "tell" Android that:

- The BroadcastReceiver named **ReceptorSMS** will become "**active**" when an "event" (**intent**) of type SMS_RECEIVED appears.

- That happens when the mobile device receives an SMS message.

The structure of the file AndroidManifest.xml

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
      package="smsTavy.com"
      android:versionCode="1"
      android:versionName="1.0">
    <uses-sdk android:minSdkVersion="8" />
    <uses-permission android:name="android.permission.SEND_SMS"></uses-permission>
    <uses-permission android:name="android.permission.RECEIVE_SMS"></uses-permission>
    <application android:icon="@drawable/icon" android:label="@string/app_name">
        <activity android:name=".AppSMStavyActivity"
                  android:label="@string/app_name">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
        <receiver android:name="ReceptorSMS">
            <intent-filter>
                <action android:name="android.provider.Telephony.SMS_RECEIVED">
                </action>
            </intent-filter>
        </receiver>
    </application>
</manifest>
```

The implementation of the application

In order to complete the application, we should complete the method **onReceive(…)** so as to obtain the phone number of the sender and the text message that makes up the message. The **onReceive(…)** has 2 parameters: a **Context** and an **Intent**.

What interests us is contained in a “packaged" form in that intent which is received as a parameter by the method☹.

The working strategy ☺

- check if our intent is of type SMS_RECEIVED;
- using the method **getExtras()**, we get the "package" (bundle) of the intent received as a parameter;
- from the "package" we obtain the list of contained objects (Object[ ]), using the **.get(“pdus")** method ;

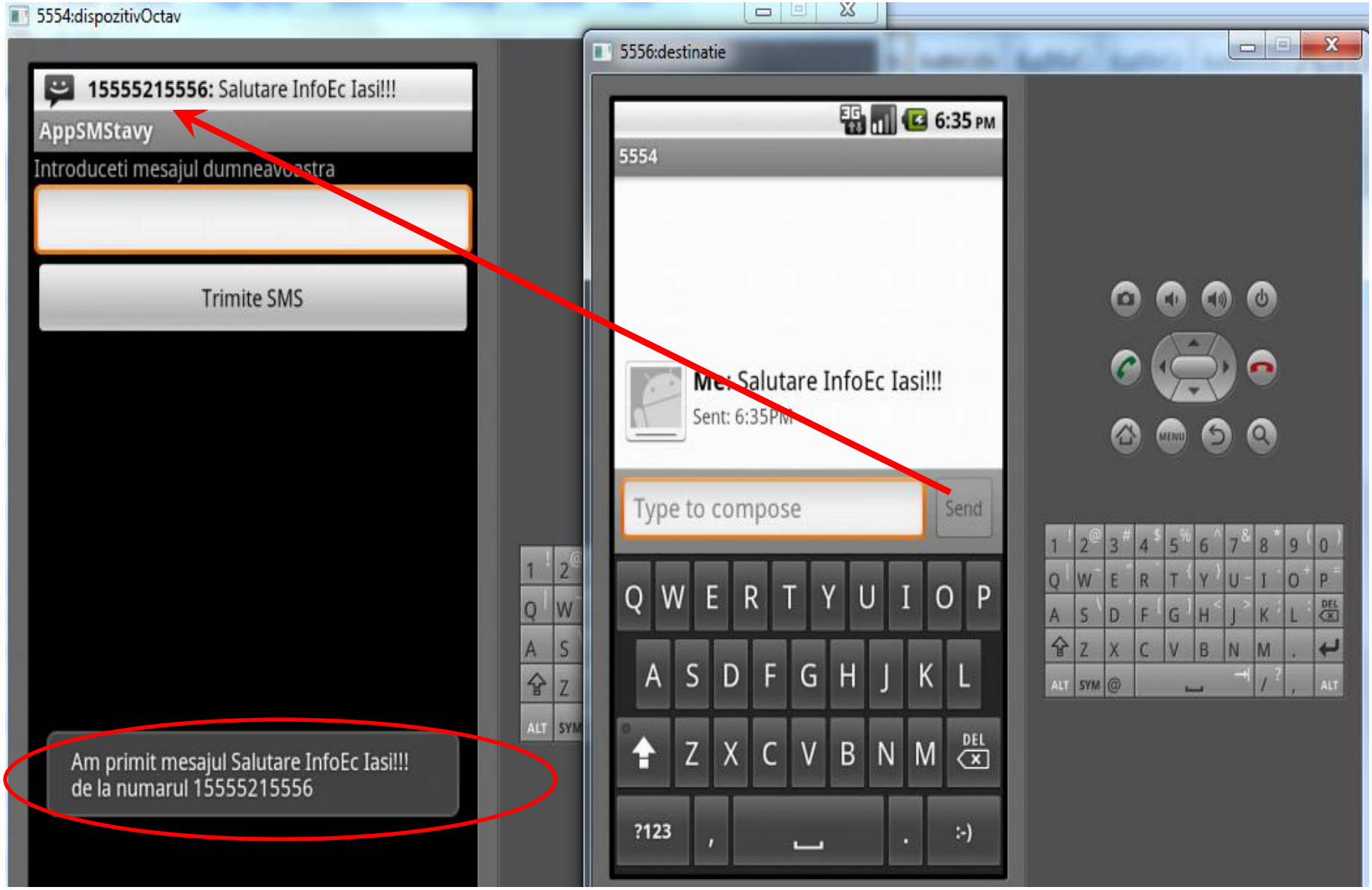
The working strategy - continuation

- define a **message** of type **SmsMessage**;
- from the **SmsMessage** class we use the method **createFromPdu** to obtain the actual message of the first object in the package;
- to get the sender's phone number and the text message, we use the methods **getOriginatingAddress** and **getMessageBody**;
- display (through a **Toast**) on the screen the obtained data.

Implementation – a model☺

```
package smsTavy.com;
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 ReceptorSMS extends BroadcastReceiver {
        @Override
        public void onReceive(Context context, Intent intent) {
                // TODO Auto-generated method stub
                Log.i("din receiver","s-a declansata metoda onReceive");
                if(intent.getAction().equals("android.provider.Telephony.SMS_RECEIVED"))
                {
                        Log.i("din receiver", "Am primit un SMS!!!");
                        Bundle pachet;
                        pachet = intent.getExtras();

                        Object[] mesajSosit =  (Object[]) pachet.get("pdus");

                        SmsMessage mesaj;
                        String format = pachet.getString("format");
                        mesaj = SmsMessage.createFromPdu((byte[]) mesajSosit[0], format);
                        String numar;
                        String text;

                        numar = mesaj.getOriginatingAddress();
                        text = mesaj.getMessageBody();

                        Toast notificare;
                        notificare = Toast.makeText(context, "Am primit mesajul " + text + " de la
numarul " + numar, Toast.LENGTH_LONG);

                        notificare.show();

                        Log.i("din receiver", numar + "   " + text);

                }
        }
}
```

Final result



Future directions

- Monitoring the children by parents☺

1. the parent sends an SMS with a special message to the child;
2. the application installed on that phone sends a reply SMS 's to the parent, providing geographical location (latitude and longitude) where the child is;
3. more jokes, more seriously, we can imagine that these locations are: school, ~~club~~☺, library etc.

The complete code (for the lab☺)

ReceptorSMS

```
package com.example.adminlocal.appcurs03;

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 ReceptorSMS extends BroadcastReceiver {

    public ReceptorSMS() {
    }

    @Override
    public void onReceive(Context context, Intent intent) {
        // TODO: This method is called when the BroadcastReceiver is receiving
        // an Intent broadcast.
        Log.i("TAVY","Am intrat cu executia in onReceive de la BroadcastReceiver");
        //aici urmeaza sa facem procesarea efectiva a intentului primit
        //verificam daca intentul este de tip mesaj SMS primit
        if(intent.getAction().equals("android.provider.Telephony.SMS_RECEIVED"))
        {
            Log.i("TAVY","Am intrat in structura if");
            Bundle pachet;
            pachet = intent.getExtras();
```

```
Object[] mesajeSosite;
        mesajeSosite = (Object[]) pachet.get("pdus");

        //in primul object avem de fapt mesajul sosit
        SmsMessage mesajPrimit;
        String format = pachet.getString("format");
        mesajPrimit = SmsMessage.createFromPdu((byte[]) mesajSosit[0], format);

        String numarExpeditor;
        String mesajExpeditor;

        numarExpeditor = mesajPrimit.getOriginatingAddress();
        mesajExpeditor=mesajPrimit.getMessageBody();
        //in acest moment avem "despachetat" mesajul primit
        //urmeaza sa valorificam informatiile primite prin afisarea lor
        //sub forma de notificare
        Toast notificare;
        notificare = Toast.makeText(context,"Mesaj primit de la " + numarExpeditor + " : " +
mesajExpeditor,Toast.LENGTH_LONG);
        notificare.show();
      }
      //throw new UnsupportedOperationException("Not yet implemented");
   }
}
```

AndroidManifest.xml

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
   package="com.example.adminlocal.appcurs03" >

   <uses-permission android:name="android.permission.RECEIVE_SMS"></uses-permission>
   <application
      android:allowBackup="true"
      android:icon="@mipmap/ic_launcher"
      android:label="@string/app_name"
      android:theme="@style/AppTheme" >
      <activity
         android:name=".FrmBlank"
         android:label="@string/app_name" >
         <intent-filter>
            <action android:name="android.intent.action.MAIN" />

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

      <receiver
         android:name=".ReceptorSMS"
         android:enabled="true"
         android:exported="true" >
         <intent-filter>
            <action android:name="android.provider.Telephony.SMS_RECEIVED">

            </action>
         </intent-filter>
      </receiver>
   </application>
```

Very important!!!

It is **mandatory** to request permissions at runtime for sending and receiving SMS messages, in onCreate() event from the activity:

```
public class MainActivity extends AppCompatActivity {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
    ActivityCompat.requestPermissions(this,new String[]{Manifest.permission.SEND_SMS, Manifest.permission.RECEIVE_SMS},1);
}
...
}
```