Envoyer un SMS

par Axon de Tuto Mobile (Tuto Mobile)

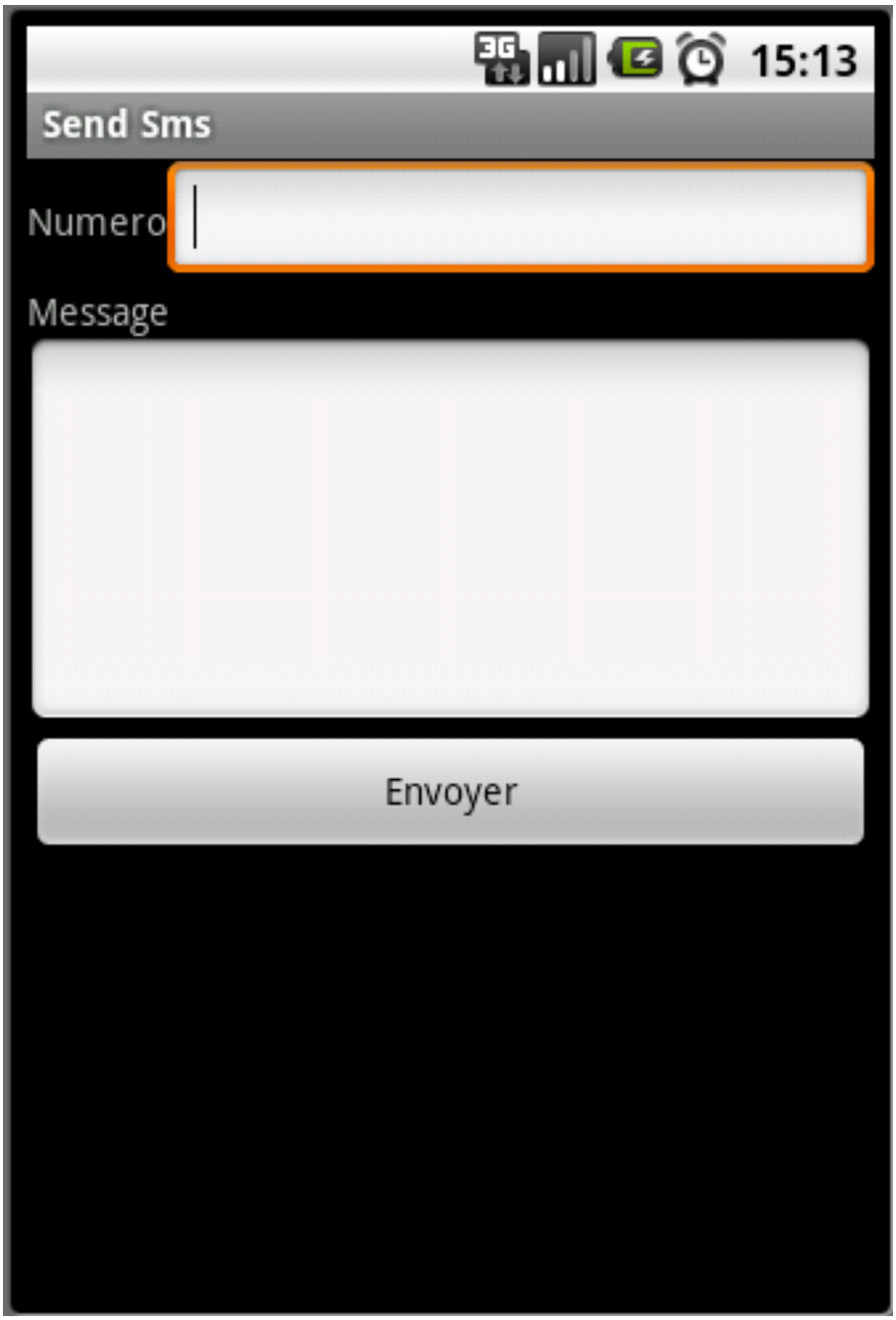
Date de publication : 26 janvier 2011

Dernière mise à jour :

Voici un nouveau tutoriel Android (depuis bien longtemps, diront les plus exigeants d'entre vous).
C'est un lecteur de Tuto Mobile qui me l'a envoyé, il s'agit de **dia100daly**, donc merci à lui.
Dans ce tutoriel nous allons apprendre à envoyer des SMS. Peut-être qu'un jour vous aurez envie de faire une application qui envoie des SMS qui sait ?
C'est toujours bien de savoir comment faire (en plus cela sera nécessaire pour le prochain tutoriel) :P Enfin trêve de bavardage.
Commençons par créer un projet avec la version 1.6 d'Android pour être conforme à tous les autres tutoriels Android du site. Pour ma part, il s'appelle *EnvoieSms* avec une activité qui porte le même nom (vous pouvez l'appeler comme vous voulez).

I - Code XML

Avant de vous donner le code XML que nous allons utiliser pour faire l'interface graphique, je vais vous montrer le rendu que l'on souhaite à la fin :



Envoi de SMS Android

Donc pour obtenir cette petite interface très simple voici le code XML correspondant :

```
<?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">
 <LinearLayout android:orientation="horizontal"
  android:layout_width="fill_parent" android:layout_height="wrap_content">
  <TextView android:layout_width="wrap_content"
```

```
    android:layout_height="wrap_content" android:text="@string/numero" />
   <EditText android:id="@+id/numero" android:layout_width="fill_parent"
    android:layout_height="wrap_content" />
  </LinearLayout>
  <TextView android:layout_width="wrap_content"
   android:layout_height="wrap_content" android:text="@string/message" />
  <EditText android:id="@+id/message" android:layout_width="fill_parent"
   android:layout_height="200sp" />
  <Button android:id="@+id/envoyer" android:layout_width="fill_parent"
   android:layout_height="wrap_content" android:text="@string/envoyer" />
</LinearLayout>
```

Pas de panique si vous avez des erreurs c'est parce que vous n'avez pas créé les ressources String. Créez les ressources qu'il faut dans le fichier *String.xml* comme ceci :

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
 <string name="numero">Numero</string>
 <string name="app_name">Envoie Sms</string>
 <string name="message">Message</string>
 <string name="envoyer">Envoyer</string>
</resources>
```

II - Code Java

Nous allons maintenant mettre en place le mécanisme d'envoi de SMS. Pour cela nous avons juste besoin d'utiliser un objet de type **SmsManager**.
Sans plus tarder éditons notre activité de départ EnvoieSms et insérer le code Java (commenté) suivant :

```
package com.sdiawara.envoiesms;

import android.app.Activity;
import android.os.Bundle;
import android.telephony.gsm.SmsManager;
import android.view.View;
import android.view.View.OnClickListener;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;

public class EnvoieSms extends Activity {

    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);
        //On récupère le bouton créé en XML grâce à son id
        Button btnEnvoie = (Button)findViewById(R.id.envoyer);
        //On récupère les deux EditText correspondant aux champs pour entrer le numéro et le message
        final EditText numero =(EditText)findViewById(R.id.numero);
        final EditText message = (EditText)findViewById(R.id.message);
        //On affecte un écouteur d'évènement au bouton
        btnEnvoie.setOnClickListener(new OnClickListener() {

   @SuppressWarnings("deprecation")
   public void onClick(View v) {
    //On récupère ce qui a été entré dans les EditText
    String num = numero.getText().toString();
    String msg = message.getText().toString();

  //Si le numéro est supérieur à 4 caractères et que le message n'est pas vide on lance la procédure d'envoi
    if(num.length()>= 4 && msg.length() > 0){

  //Grâce à l'objet de gestion de SMS (SmsManager) que l'on récupère via la méthode static getDefault()
     //On envoie le SMS à l'aide de la méthode sendTextMessage
     SmsManager.getDefault().sendTextMessage(num, null, msg, null, null);
```

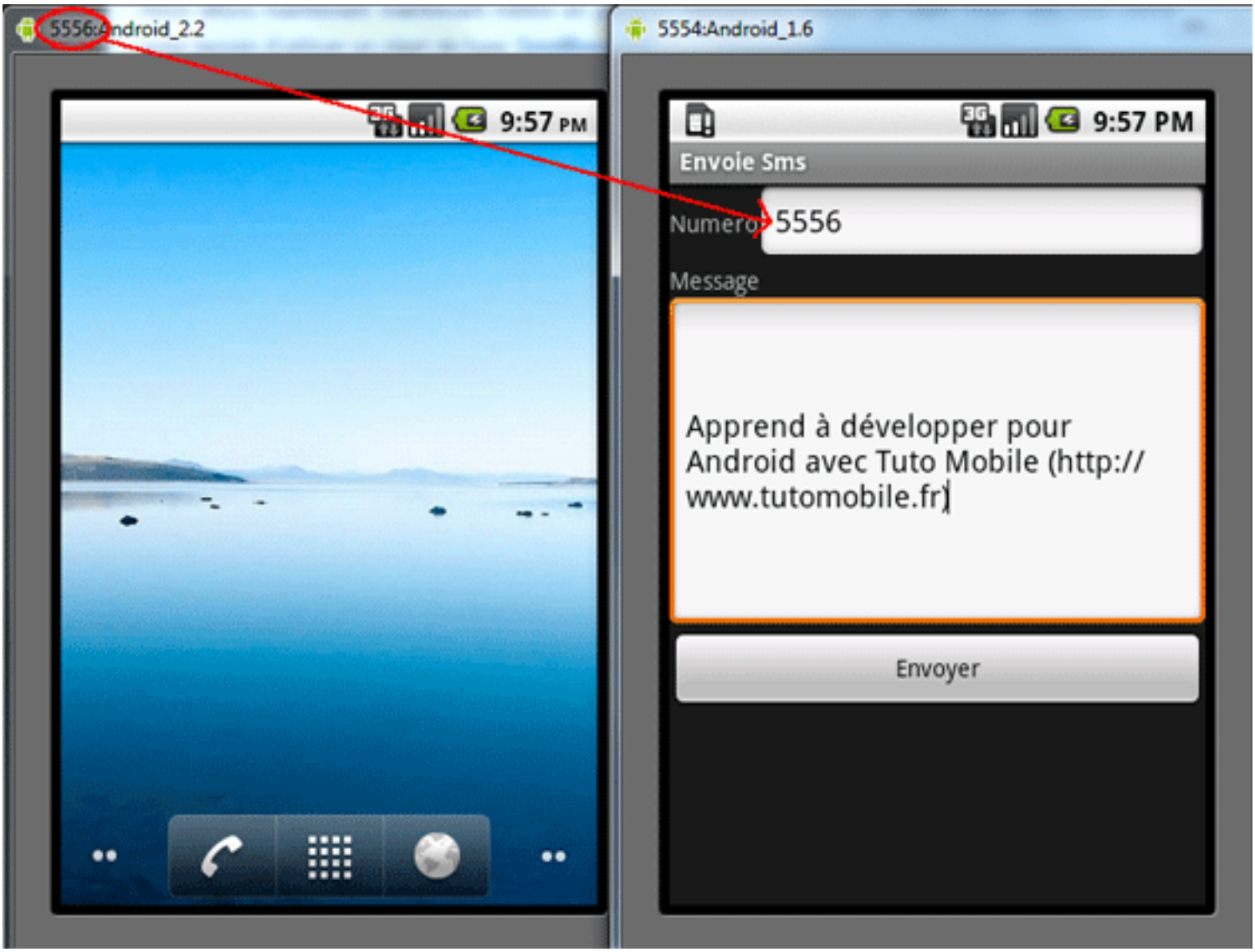
```
      //On efface les deux EditText
      numero.setText("");
      message.setText("");
     }else{
      //On affiche un petit message d'erreur dans un Toast
      Toast.makeText(EnvoieSms.this, "Enter le numero et/ou le message", Toast.LENGTH_SHORT).show();
     }

    }
   });
     }
}
```

III - AndroidManifest.xml

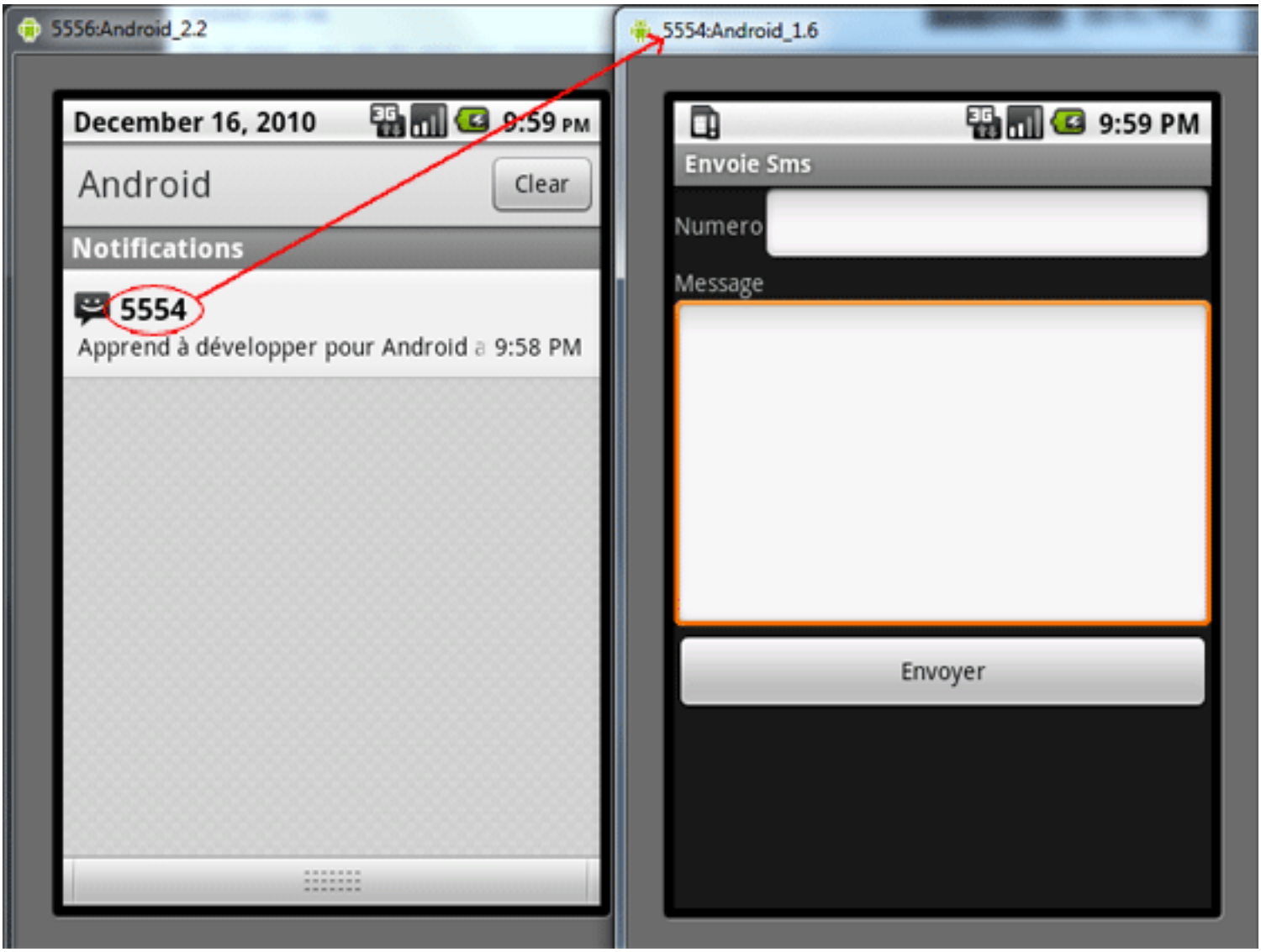
Voilà nous avons presque fini il nous reste juste à demander la permission d'envoyer des messages. Pour cela, ouvrez le fichier *AndroidManifest.xml* et ajoutez cette ligne :

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

Vous pouvez maintenant tester. Pour bien voir que ça marche nous allons utiliser deux émulateurs. Je suppose que vous en disposez déjà d'un, nous allons donc en créer un deuxième. Cliquez sur le menu **Windows >>> Android SDK and AVD Manager**. Ensuite cliquez sur **New** pour créer un nouvel émulateur, donnez-lui le nom que vous voulez, cliquez sur **Create AVD** et sélectionnez ce nouvel émulateur puis cliquez sur **Start**. Il ne reste plus qu'à lancer votre projet avec l'un des émulateurs et à rentrer le numéro qui apparaît sur l'autre émulateur comme numéro du destinataire. Voilà tout devrait marcher normalement !



envoi sms android



envoi sms android

A bientôt pour un prochain tutoriel.

IV - Lien

Envoyer un sms depuis Android (tutomobile)