# 收发短信

## 一、发短信

### 1.1发短信的步骤：

1.1.1、使用SmsManager.getDefault()函数获得一个SmsManager实例。

SmsManager smsManager=SmsManager.*getDefault*();

1.1.2、使用SmsManager实例的sendTextMessage来发送短信

smsManager.sendTextMessage(to.getText().toString(), **null**, msgInput.getText().toString(), null, **null**);

1.1.3、声明发送短信权限：

<uses-permission android:name=*"android.permission.SEND\_SMS"*/>

1.1.4、为了监听短信发送状态，可在发送短信时将一个PendingIntent意图传入，当短信发送完毕时，系统会广播该意图，注册一个广播接收器来接收该广播，并在广播接收器中根据返回值，判断短信发送状态。

注册一个广播接收器的步骤如下：

1）、定义一个派生于BroadcastReceiver的广播接收器类

2）、定义一个广播过滤器IntentFilter和一个广播接收器子类对象

3）、使用IntentFilter的addAction方法来设置要监听的广播

4）、使用registerReceiver来注册广播

### 1.2 关键代码

**protected** **void** onCreate(Bundle savedInstanceState) {

**super**.onCreate(savedInstanceState);

setContentView(R.layout.*activity\_main*);

to=(EditText)findViewById(R.id.*to*);

msgInput=(EditText)findViewById(R.id.*msg\_input*);

sendFilter=**new** IntentFilter();

sendFilter.addAction("SEND\_ACTION");

statusReceiver=**new** SendStatusReceiver();

registerReceiver(statusReceiver, sendFilter);

send=(Button)findViewById(R.id.*send*);

send.setOnClickListener(**new** OnClickListener() {

@Override

**public** **void** onClick(View v) {

SmsManager smsManager=SmsManager.*getDefault*();

Intent sendIntent=**new** Intent("SEND\_ACTION");

PendingIntent pi=PendingIntent.*getBroadcast*(MainActivity.**this**, 0, sendIntent, 0);

smsManager.sendTextMessage(to.getText().toString(),

**null**, msgInput.getText().toString(), pi, **null**);

}

});

}

@Override

**protected** **void** onDestroy() {

// **TODO** 自动生成的方法存根

**super**.onDestroy();

unregisterReceiver(statusReceiver);

}

**public** **class** SendStatusReceiver **extends** BroadcastReceiver{

@Override

**public** **void** onReceive(Context context, Intent intent) {

**if**(getResultCode()==*RESULT\_OK*)

{

Toast.*makeText*(context, "Send succeeded", Toast.*LENGTH\_SHORT*).show();

}

**else**

{

Toast.*makeText*(context, "Send failed", Toast.*LENGTH\_SHORT*).show();

}

}

}

注意，Intent sendIntent=**new** Intent("SEND\_ACTION");是创建一个Intent，并设置Action为”SEND\_ACTION”，这里的”SEND\_ACTION”是程序员自定义的，并不是系统预定义的，因此，也可以改为其它的值，但要与广播接收器中过滤器设置监听的ACTION相同。

## 二、收短信

当手机接收到一条短信的时候，系统会发出一条android.provider.Telephony.SMS\_RECEIVED的广播，这条广播里会携带与短信相关的所有数据，应用程序可以监听该广播，并解析出短信的内容。注册一个广播接收器的方法与1.1.4中相同。

派生一个广播接口器类：

**public** **class** MessageReceiver **extends** BroadcastReceiver{

String address;

String fullMessage;

@Override

**public** **void** onReceive(Context context, Intent intent) {

Bundle bundle=intent.getExtras();

Object[] pdus=(Object[])bundle.get("pdus");//提取短信消息

SmsMessage[] messages=**new** SmsMessage[pdus.length];

**for**(**int** i=0;i<messages.length;i++)

{

messages[i]=SmsMessage.*createFromPdu*((**byte**[])pdus[i]);

}

address=messages[0].getOriginatingAddress();

fullMessage="";

**for**(SmsMessage message:messages)

{

fullMessage+=message.getMessageBody();

}

sender.setText(address);

content.setText(fullMessage);

abortBroadcast();

}

}

其中，Bundle类是一个key-value对。pdu为承载着一条短信的所有短信。一条短信为140个英文字符长度，在这个长度范围内，即需一个pdu即可。超出这个范围，即要分割成多个pdu数组。

注册一个广播：

//广播过滤器

receiverFilter=**new** IntentFilter();

receiverFilter.setPriority(100); receiverFilter.addAction("android.provider.Telephony.SMS\_RECEIVED");

//使用设置好的广播过滤器来创建一个广播接收器

messageReceiver=**new** MessageReceiver();

//注册接收器

registerReceiver(messageReceiver, receiverFilter);

记得在Acivity的OnDestroy函数中对广播进行反注册：

unregisterReceiver(messageReceiver);