Android程式設計

簡訊處理

王昱景 Brian Wang
brian.wang.frontline@gmail.com

- 行動裝置的主要功能就是對外通訊，除了使用語音通話外，另一個常用功能是簡訊 (Short Message Service, SMS) 即手機的文字訊息服務
- 在 Android 應用程式收發簡訊需要使用廣播接收器來取得與顯示簡訊的內容

SMSDemo

- 開啟和執行 Android 專案
- 建立寄送簡訊使用介面的版面配置
- 建立 Activity 活動類別寄送簡訊
- 建立 BroadcastReceiver 類別接收簡訊
- 在 AndroidManifest.xml 註冊廣播接收器和新增權限

1.開啟和執行Android專案

- 請啟動 Eclipse IDE
- 建立 Android 專案
  - Project Name: SMSDemo
  - Build Target: Android API 4.2.2
  - Package Name

2.建立寄送簡訊使用介面的版面配置

- 使用 LinearLayout 垂直編排 2 個 TextView 和 EditText 元件，一個 Button 元件

```
<TextView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="@string/lab1" />

    <EditText
        android:id="@+id/txtPhoneNo"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:inputType="phone" />

    <TextView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="@string/lab2" />

    <EditText
        android:id="@+id/txtMessage"
        android:layout_width="match_parent"
        android:layout_height="150dp"
        android:gravity="top"
        android:inputType="text" />

    <Button
        android:id="@+id/btnSendSMS"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:onClick="btnSendSMS_Click"
        android:text="@string/btnSendSMS" />
```

3. 建立 Activity 活動類別寄送簡訊

- 在活動類別的開頭宣告成員變數 EditText 物件 txtPhoneNo 和 txtMessage

```
public class MainActivity extends Activity {

   private EditText txtPhoneNo, txtMessage;

}
```

onCreate() 方法

- 在覆寫的 onCreate() 方法載入版面配置後，可以取得 2 個 EditText 物件

```
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);

    // 取得EditText元件
    txtPhoneNo = (EditText) findViewById(R.id.txtPhoneNo);
    txtMessage = (EditText) findViewById(R.id.txtMessage);
}
```

btnSendSMS_Click() 事件處理方法

- 在 Button 元件的事件處理方法寄送簡訊
- 首先取得簡訊目標的電話號碼與簡訊內容

```
// Button元件的事件處理 - 寄送簡訊
public void btnSendSMS_Click(View view) {
    // 取得簡訊內容
    String phoneNo = txtPhoneNo.getText().toString();
    String message = txtMessage.getText().toString();

    if (phoneNo.length() > 0 && message.length() > 0) {
        sendSMS(phoneNo, message); // 送出簡訊
    } else {
        Toast.makeText(this, "請確認輸入電話號碼和訊息內容!", Toast.LENGTH_SHORT).show();
    }
}
```

sendSMS() 方法

- 在自訂 sendSMS() 方法寄送簡訊
- 為了知道簡訊是否順利送達，註冊 2 個廣播接收器來取得簡訊的傳送結果

```java
// 寄送簡訊
private void sendSMS(String phoneNumber, String message) {
	String SENT = "SMS_SENT";
	String DELIVERED = "SMS_DELIVERED";

	// 當簡訊已經送出, 建立廣播接收器來取得結果
	registerReceiver(new BroadcastReceiver() {
		@Override
		public void onReceive(Context content, Intent intent) {
			switch (getResultCode()) {
				case Activity.RESULT_OK:
					Toast.makeText(getBaseContext(), "簡訊送出", Toast.LENGTH_SHORT).show();
					break;
				case SmsManager.RESULT_ERROR_GENERIC_FAILURE:
					Toast.makeText(getBaseContext(), "一般錯誤!", Toast.LENGTH_SHORT).show();
					break;
				case SmsManager.RESULT_ERROR_NO_SERVICE:
					Toast.makeText(getBaseContext(), "沒有服務!", Toast.LENGTH_SHORT).show();
					break;
				case SmsManager.RESULT_ERROR_NULL_PDU:
					Toast.makeText(getBaseContext(), "空的PDU", Toast.LENGTH_SHORT).show();
					break;
				case SmsManager.RESULT_ERROR_RADIO_OFF:
					Toast.makeText(getBaseContext(), "沒有訊號", Toast.LENGTH_SHORT).show();
					break;
			}
		}
	}, new IntentFilter(SENT));
	// 當簡訊已經送達, 建立廣播接收器來取得結果
	registerReceiver(new BroadcastReceiver() {
		@Override
		public void onReceive(Context content, Intent intent) {
			switch (getResultCode()) {
				case Activity.RESULT_OK:
					Toast.makeText(getBaseContext(), "簡訊已經送達!", Toast.LENGTH_SHORT).show();
					break;
				case Activity.RESULT_CANCELED:
					Toast.makeText(getBaseContext(), "簡訊沒有送達!", Toast.LENGTH_SHORT).show();
					break;
			}
		}
	}, new IntentFilter(DELIVERED));

	// 建立PendingIntent物件
	PendingIntent sentPI = PendingIntent.getBroadcast(this, 0, new Intent(SENT), 0);
	PendingIntent deliveredPI = PendingIntent.getBroadcast(this, 0, new Intent(DELIVERED), 0);

	// 取得SmsManager物件
	SmsManager sms = SmsManager.getDefault();
	// 送出簡訊
	sms.sendTextMessage(phoneNumber, null, message, sentPI, deliveredPI);
}
```

4. 建立 BroadcastReceiver 類別接收簡訊

- 在 Android 應用程式可以使用廣播接收器接收行動裝置收到簡訊的系統廣播
- 可以透過它來過濾出需要的簡訊

```java
public class SMSReceiver extends BroadcastReceiver {

    @Override
    public void onReceive(Context context, Intent intent) {
        // 取得收到的簡訊內容
        Bundle bundle = intent.getExtras();
        SmsMessage[] msgs = null;
        String str = "";

        // 如果有內容
        if (bundle != null) {
            // 取出訊息內容
            Object[] pdus = (Object[]) bundle.get("pdus");
            msgs = new SmsMessage[pdus.length];

            for (int i = 0; i < msgs.length; i++) {
                msgs[i] = SmsMessage.createFromPdu((byte[]) pdus[i]);
                str += "SMS from " + msgs[i].getOriginatingAddress();
                str += " :";
                str += msgs[i].getMessageBody().toString();
                str += "\n";
            }

            // 顯示取得的訊息內容
            Toast.makeText(context, str, Toast.LENGTH_LONG).show();
        }
    }

}
```

5. 在 AndroidManifest.xml 註冊廣播接收器和新增權限

- SMSReceiver 廣播接收器需要在 AndroidManifest.xml 檔註冊
- Telephony.SMS_RECEIVED 是處理行動裝置收到簡訊的系統廣播

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

```
        <receiver android:name="SMSReceiver" >
            <intent-filter>
                <action android:name="android.provider.Telephony.SMS_RECEIVED" />
            </intent-filter>
        </receiver>
```