广播事件

广播事件

- 所有的事件列表见：http://developer.android.com/reference/android/content/Intent.html#ACTION_PICK
- android.os.action.POWER_SAVE_MODE_CHANGED
- android.provider.Telephony.SIM_FULL
- android.provider.Telephony.SMS_CB_RECEIVED
- android.provider.Telephony.SMS_DELIVER
- android.provider.Telephony.SMS_EMERGENCY_CB_RECEIVED
- android.provider.Telephony.SMS_RECEIVED
- android.provider.Telephony.SMS_REJECTED
- android.provider.Telephony.SMS_SERVICE_CATEGORY_PROGRAM_DATA_RECEIVED
- android.provider.Telephony.WAP_PUSH_DELIVER
- android.provider.Telephony.WAP_PUSH_RECEIVED
- 。。。

注册事件

- 1)静态注册：在AndroidManifest.xml注册，android不能自动销毁广播接收器，也就是说当应用程序关闭后，还是会接收广播。

- 2)动态注册：在代码中通过registerReceiver()手工注册.当程序关闭时,该接收器也会随之销毁。当然，也可手工调用unregisterReceiver()进行销毁。

示例代码

- 静态
- <application>
- <activity name=""/>
- <receiver android:name=".MyBroadcastReceiver">
- <!-- intent过滤器,指定可以匹配哪些intent, 一般需要定义action 可以是自定义的也可是系统的 -->
- <intent-filter>
- <action android:name="com.app.bc.test"/>
- </intent-filter>
- </receiver>
- </application>

示例代码

- 动态注册
- //生成一个BroadcastReceiver对象
- SMSReceiver smsReceiver = new SMSReceiver();
- //生成一个IntentFilter对象
- IntentFilter filter = new IntentFilter();
- filter.addAction(“android.provider.Telephony.SMS_RECEIVED”);
- //将BroadcastReceiver对象注册到系统当中
- //此处表示该接收器会处理短信事件
- TestBC1Activity.this.registerReceiver(smsReceiver, filter);

代码示例

- 处理逻辑
- class MyBroadcastReceiver extends BroadcastReceiver {
- 　　//接收到广播会被自动调用
- 　　@Override
- 　　public void onReceive (Context context, Intent intent) {
- 　　　　//从Intent中获取action
- 　　　　…your code here…
- 　　}
- }

参考资料

- http://blog.csdn.net/luoshengyang/article/details/6730748

- http://drops.wooyun.org/tips/4393