

# 事件处理

## 1.1 知识解析

使用 UI 事件来实现和用户的互动：在 Android 中有多种监听用户与应用程序互动的方法。

当事件发生在用户接口上时，截获使用者与特定 View 互动的事件，在 View 类中就提供了此类的方法。

例如：当一个 View (按钮) 被点击，这时 `onTouchEvent()` 方法就会被调用。

为了监听这个动作，必需继承对应的类并覆盖这个方法，那么，为了处理这类事件，每个 View 都需要被继承，实际上来说并不实用。

View 类也包含嵌套接口的集合，使得这些事件可以更容易去定义，这些接口就被称为事件监听器。

事件监听器一般定义为 View 类中内部接口，它包含若干回调方法，在事件发生的时候这些方法会被调用。

当使用者与 View 的对象互动，此 listener 需要被注册，这些回调方法将会由 Android framework 所调用。

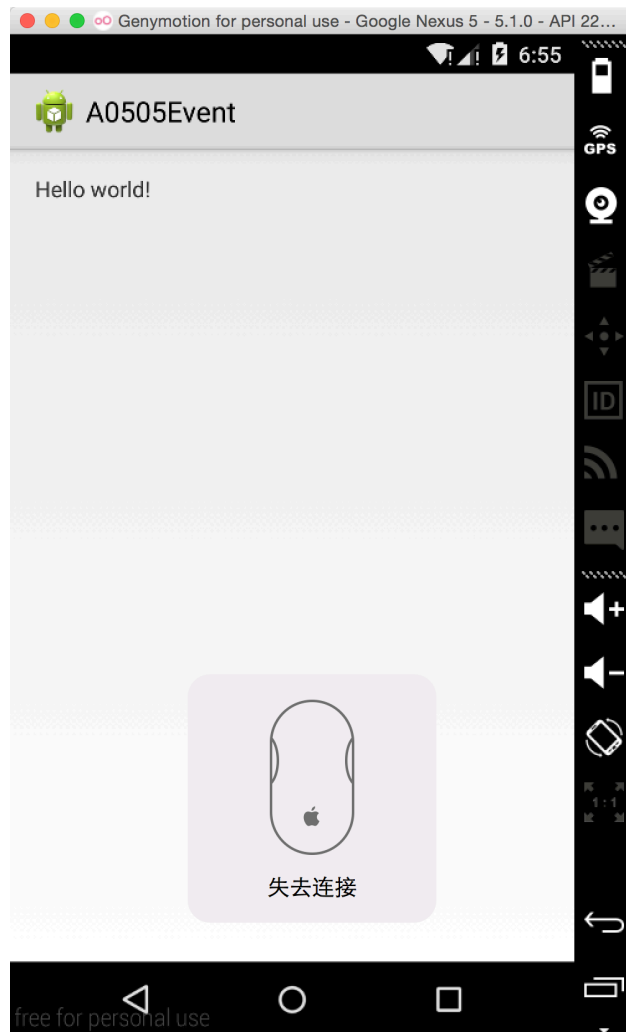
事件监听器的不同类型回调方法：

- `onClick()`：定义在 `View.OnClickListener`
- `onLongClick()`：定义在 `View.OnLongClickListener`
- `onFocusChange()`：定义在 `View.OnFocusChangeListener`
- `onKey()`：定义在 `View.OnKeyListener`
- `onTouch()`：定义在 `View.OnTouchListener`

`onClick()` 没有回传值，但是一些其它监听器方法必需回传一个布尔值。它返回的布尔值表明你已经完成了这个事件的处理，还是应该把它继续传下去。返回 `true` 表明已经

处理完成并且停止了传递，如果返回为 `false` 表明事件还没有完成，或者它还需要继续被传递给其他的监听器。

## 1.2 功能演示



## 1.3 实战操作

```
tv.setOnClickListener(new OnClickListener() {  
  
    @Override  
    public void onClick(View v) {
```

```
// TODO Auto-generated method stub
Log.i("OnClick", "OnClick");
}
});
tv.setOnLongClickListener(new OnLongClickListener() {

    @Override
    public boolean onLongClick(View v) {
        // TODO Auto-generated method stub
        Log.i("LongClick", "LongClick");
        return false;
    }
});
tv.setOnTouchListener(new OnTouchListener(){

    @Override
    public boolean onTouch(View v, MotionEvent event) {
        // TODO Auto-generated method stub
        Log.i("ONTouch", "onTouch:"+event.getAction());
        v.performClick();
        return true;
    }
});
```

## 1.4 职业素质

使用 UI 事件来实现和用户的互动：在 Android 中有多种监听用户与应用程序互动的方法。

当事件发生在用户接口上时，截获使用者与特定 View 互动的事件，在 View 类中就提供了此类的方法。