



链滴

Android EditText 响应 drawableRight 等的点击事件

作者: [Single](#)

原文链接: <https://ld246.com/article/1500878889413>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

我们在输入框EditText中，经常会添加比如删除X，下拉箭头等图标，以得到更好的用户体验和需求。如果放在右边，我们经常设置drawableRight属性来实现，那么这些图标如何响应点击事件呢？下面记录一下！

方法

首先我们要继承AppCompatActivity，实现一个自定义的EditText（现在一般为了兼容和实现MD风，都会继承AppCompatActivity，而不再去继承EditText，另外AS也会给出相应的错误警告提示！）

然后借助于onTouchEvent，根据触摸位置来响应图标的点击事件，判断手指抬起的时候的x, y坐标是否点击在drawable对象上。其中需要搞清楚这几个参数：

```
event.getRawX()//相对于左边界的绝对坐标，以左上角为 (0,0)
event.getX()//相对于自身的坐标，以该空间的左上角为 (0,0)
getLeft()//相当于margin,控件左边界相对于父控件的距离
getPaddingLeft()//相当于padding, 控件中元素相对于控件的间距
getBounds().width()//获取元素绘制区域的宽度
drawableRight.getIntrinsicWidth()//获取drawable的实际宽度
```

那么就需要得到这个图标，这里通过调用getCompoundDrawables() 可以获取一个长度为4的drawable数组，存放drawableLeft, Right, Top, Bottom四个图片资源对象：

```
final int DRAWABLE_LEFT = 0;
final int DRAWABLE_TOP = 1;
final int DRAWABLE_RIGHT = 2;
final int DRAWABLE_BOTTOM = 3;
```

可以根据图标的位置，拿到对应的图片Drawable。

最后，通过声明一个接口，定义一个回调方法，这里就不细说了，比较常见了！

```
public class DropDownEditText extends AppCompatActivity {
    final int DRAWABLE_LEFT = 0;
    final int DRAWABLE_TOP = 1;
    final int DRAWABLE_RIGHT = 2;
    final int DRAWABLE_BOTTOM = 3;

    public DropDownEditText(Context context) {
        super(context);
    }

    public DropDownEditText(Context context, AttributeSet attrs) {
        super(context, attrs);
    }

    public interface OnDropArrowClickListener {
        void onDropArrowClick();
    }

    private OnDropArrowClickListener onDropArrowClickListener;

    public void setOnDropArrowClickListener(OnDropArrowClickListener onDropArrowClickList
```

```

ner) {
    this.onDropArrowClickListener = onDropArrowClickListener;
}

@Override
public boolean onTouchEvent(MotionEvent event) {

    if (event.getAction() == MotionEvent.ACTION_UP) {
        Drawable drawableRight = getCompoundDrawables()[DRAWABLE_RIGHT];
        if (drawableRight != null) {
            //本次点击事件的x轴坐标, 如果>当前控件宽度-控件右间距-drawable实际展示大小
            if (event.getX() >= (getWidth() - getPaddingRight() - drawableRight.getIntrinsicWid
h())) {

                //设置点击EditText右侧图标EditText失去焦点,
                //防止点击EditText右侧图标EditText获得焦点, 软键盘弹出
                setFocusableInTouchMode(false);
                setFocusable(false);
                if (onDropArrowClickListener != null) {
                    onDropArrowClickListener.onDropArrowClick();
                }
            } else {
                setFocusableInTouchMode(true);
                setFocusable(true);
            }
        }
    }
    return super.onTouchEvent(event);
}
}

```

示例中展示了图标在右边的情况, 同样的其他的位置:

```

**右边: **
//另外一种直接的方法
//其实就是算该drawable最左边的x坐标,drawableRight.getIntrinsicWidth()==drawableRight.g
tBounds().width()等于图标的宽度
event.getRawX() >= (getRight() - drawableRight.getBounds().width())
**左边: **

/**getX是相对于控件本身的左上角的x坐标,<= 控件左边距+图片对象实际的宽度.这边的getLeft
当于margin,getPaddingLeft相当于padding*/
event.getX() <= getLeft() + drawableLeft.getIntrinsicWidth()

event.getRawX() <= (getLeft() + drawableLeft.getBounds().width())

**上边: **
event.getY() <= getTop() + drawableTop.getIntrinsicHeight()

**下边: **
event.getY() > getBottom() - drawableBottom.getIntrinsicHeight()

```