

Android Scroller用法

发布于2019-10-22 14:22:20 阅读 625

Android里 **Scroller** 类是为了实现View平滑滚动的一个 **Helper** 类。

通常在自定义的View时使用，在View中定义一个私有成员mScroller = new Scroller(context)。

设置mScroller滚动的位置时，**并不会导致View的滚动**，通常是用mScroller记录/计算View滚动的位置，再重写View的computeScroll()，完成实际的滚动。

相关API介绍如下

```
1 mScroller.getCurrX() //获取mScroller当前水平滚动的位置
2 mScroller.getCurrY() //获取mScroller当前竖直滚动的位置
3 mScroller.getFinalX() //获取mScroller最终停止的水平位置
4 mScroller.getFinalY() //获取mScroller最终停止的竖直位置
5 mScroller.setFinalX(int newX) //设置mScroller最终停留的水平位置，没有动画效果，直接跳到目标位置
6 mScroller.setFinalY(int newY) //设置mScroller最终停留的竖直位置，没有动画效果，直接跳到目标位置
7
8 //滚动，startX, startY为开始滚动的位置，dx,dy为滚动的偏移量，duration为完成滚动的时间
9 mScroller.startScroll(int startX, int startY, int dx, int dy) //使用默认完成时间250ms
10 mScroller.startScroll(int startX, int startY, int dx, int dy, int duration)
11
12 mScroller.computeScrollOffset() //返回值为boolean，true说明滚动尚未完成，false说明滚动已经完成。这是一
```

例子

举例说明，自定义一个CustomView，使用Scroller实现滚动

```
1 import android.content.Context;
2 import android.util.AttributeSet;
3 import android.util.Log;
4 import android.view.View;
5 import android.widget.LinearLayout;
6 import android.widget.Scroller;
7
8 public class CustomView extends LinearLayout {
9     private static final String TAG = "Scroller";
10    private Scroller mScroller;
11    public CustomView(Context context, AttributeSet attrs) {
12        super(context, attrs);
13        mScroller = new Scroller(context);
14    }
15
16    //调用此方法滚动到目标位置
17    public void smoothScrollTo(int fx, int fy) {
18        int dx = fx - mScroller.getFinalX();
19        int dy = fy - mScroller.getFinalY();
20        smoothScrollBy(dx, dy);
21    }
22
23    //调用此方法设置滚动的相对偏移
24    public void smoothScrollBy(int dx, int dy) {
25        //设置mScroller的滚动偏移量
26        mScroller.startScroll(mScroller.getFinalX(), mScroller.getFinalY(), dx, dy);
27        invalidate(); //这里必须调用invalidate()才能保证computeScroll()会被调用，否则不一定会刷新界面，
28    }
29
30    @Override
31    public void computeScroll() {
```

```
32 | //先判断mScroller滚动是否完成
33 | if (mScroller.computeScrollOffset()) {
34 |     //这里调用View的scrollTo()完成实际的滚动
35 |     scrollTo(mScroller.getCurrX(), mScroller.getCurrY());
36 |     //必须调用该方法，否则不一定能看到滚动效果
37 |     postInvalidate();
38 | }
39 | super.computeScroll();
40 | }
41 | }
```

invalidate和postInvalidate区别

两者作用一样都是 通知UI刷新 ，但是 `invalidate` 只能在 `UI线程` 中调用，`postInvalidate` 则在 `非UI线程` 中调用

scrollTo

`scrollTo(x,y)`是View的方法，其中x,y参数可以理解为View的左上角坐标从(0,0)移动到(x,y)

举例说明 `scrollTo(-5,10)` 就是 向右移动5` 向下移动10`

JavaScript

举报