

Android TextView 前面添加标签

林生

引用

醉过才知酒浓，爱过才知情重。你不能做我的诗，正如我不能做你的梦。

落寞的孤辰,乱了谁的发梢,三生的河畔,许了谁的流年。奈何桥上谁饮了那孟婆汤刻下这遥远的怀念,遗忘,感伤!月老手中错绑的红线,苍涩了千年。似水流年,允我相思不尽;蝶花陨落,许你一世柔情;残缺了月圆,凄美了誓言,终是那一世花开,这毕生花落。*

突然想玩TextView前面加标签

- 思路
 - 通过html
 - 通过span
- 代码实现：
 - 自定义TextView
 - 自定义一个标准的tag布局：
 - 具体使用（xml）
 - 具体使用（java）

心里想法：

好好的textView前面加了个标签...
这要是单行还好，直接一个drawable了事
但是多行...我可以自定义一个ShapeDrawable，实现drawableLeft,但是有一个问题，如果是这样的

实现：

通过html

最简单的方法肯定是通过html，不用计算什么start，end，而且现成就有的样式都可以拿：
html样式代码如下：
`I love android`
然而...并不支持，貌似是textView不支持这个border-radius属性...而我贫乏的html知识也只能记得这

通过span

这个是最靠谱的解决方法了吧，其实span可以帮助我们解决几乎所有关于textview的问题，可是貌似具体可以[了解下](https://blog.csdn.net/qq_16430735/article/details/50427978)。

什么是SpannableString?

SpannableString，是CharSequence的一种，原本的CharSequence只是一串字符序列，没有任何样式，创建方式：

```
SpannableString spannableString = new SpannableString("我是小阿曼");
```

如何对SpannableString进行润饰？

这里讲解一下几个参数的意义

what：对SpannableString进行润色的各种Span；

int：需要润色文字段开始的下标；

end：需要润色文字段结束的下标；

flags：决定开始和结束下标是否包含的标志位，有四个参数可选

SPAN_INCLUSIVE_EXCLUSIVE：包括开始下标，但不包括结束下标

SPAN_EXCLUSIVE_INCLUSIVE：不包括开始下标，但包括结束下标

SPAN_INCLUSIVE_INCLUSIVE：既包括开始下标，又包括结束下标

SPAN_EXCLUSIVE_EXCLUSIVE：不包括开始下标，也不包括结束下标

这里涉及到一个重要的角色，就是各种各样的span，它决定我们要对文字的进行怎样的润饰，

代码如下：

自定义TextView：自定义了TextView在其内部进行处理，免得每次都是一套function方法复制来！

```
package com.example.ly.testdrawable;
import android.content.Context;
import android.graphics.Bitmap;
import android.graphics.drawable.BitmapDrawable;
import android.graphics.drawable.Drawable;
import android.text.Spannable;
import android.text.SpannableString;
import android.text.style.ImageSpan;
import android.util.AttributeSet;
import android.util.Log;
import android.view.Gravity;
import android.view.LayoutInflater;
import android.view.View;
import android.widget.TextView;
import java.util.List;

/**
 * Author: linsheng
 * Data: 2018/12/10-23:55
 * Description:
 */
public class TagTextView extends android.support.v7.widget.AppCompatTextView {

    private StringBuffer content_buffer;

    private TextView tv_tag;

    private View view;//标签布局的最外层布局

    private Context mContext;
```

//必须重写所有的构造器，否则可能会出现无法inflate布局的错误！

```

public TagTextView(Context context) {

    super(context);

    mContext = context;

}

public TagTextView(Context context, AttributeSet attrs) {

    super(context, attrs);

    mContext = context;

}

public TagTextView(Context context, AttributeSet attrs, int defStyleAttr) {

    super(context, attrs, defStyleAttr);

    mContext = context;

}

public void setContentAndTag(String content, List<String> tags) {
    content_buffer = new StringBuffer();
    for (String item : tags) { // 将每个tag的内容添加到content后边，之后将用drawable替代
        content_buffer.append(item);
    }
    content_buffer.append(content);
    SpannableString spannableString = new SpannableString(content_buffer);
    for (int i = 0; i < tags.size(); i++) {
        String item = tags.get(i);
        View view = LayoutInflater.from(mContext).inflate(R.layout.tag, null); // R.layout.tag是
        tv_tag = view.findViewById(R.id.tv_tag);
        tv_tag.setText(item);
        Bitmap bitmap = convertViewToBitmap(view);
        Drawable d = new BitmapDrawable(bitmap);
        d.setBounds(0, 0, tv_tag.getWidth(), tv_tag.getHeight()); // 缺少这句话，不会报错
        ImageSpan span = new ImageSpan(d, ImageSpan.ALIGN_BOTTOM); // 图片将对齐底部边线
        int startIndex;
        int endIndex;
        startIndex = getLastLength(tags, i);
        endIndex = startIndex + item.length();
        Log.e("tag", "the start is" + startIndex + "the end is" + endIndex);
        spannableString.setSpan(span, startIndex, endIndex, Spannable.SPAN_EXCLUSIVE_EXCLUSIV
    }
    setText(spannableString);
    setGravity(Gravity.CENTER_VERTICAL);
}

private static Bitmap convertViewToBitmap(View view) {

    view.measure(View.MeasureSpec.makeMeasureSpec(0, View.MeasureSpec.UNSPECIFIED), View.Meas

    view.layout(0, 0, view.getMeasuredWidth(), view.getMeasuredHeight());

```

```

        view.buildDrawingCache();

        Bitmap bitmap = view.getDrawingCache();

        return bitmap;
    }

    private int getLastLength(List<String> list, int maxLength) {
        int length = 0;
        for (int i = 0; i < maxLength; i++) {
            length += list.get(i).length();
        }
        return length;
    }
}

```

自定义一个标准的tag布局：

```

<?xml version="1.0" encoding="utf-8"?>

<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical">
    <TextView
        android:id="@+id/tv_tag"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="center_vertical"
        android:layout_marginBottom="2dp"
        android:layout_marginRight="5dp"
        android:background="@drawable/bg_blue_20"
        android:gravity="center_vertical"
        android:paddingBottom="2dp"
        android:paddingLeft="3dp"
        android:paddingRight="3dp"
        android:text="标签"
        android:textColor="@android:color/white"
        android:textSize="12sp"/>
</LinearLayout>

```

具体使用（xml）：

```

<?xml version="1.0" encoding="utf-8"?>
<android.support.constraint.ConstraintLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context="com.example.ly.testdrawable.MainActivity">

    <com.example.ly.testdrawable.TagTextView
        android:id="@+id/text"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Hello World!"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintLeft_toLeftOf="parent"
        app:layout_constraintRight_toRightOf="parent"

```

```
app:layout_constraintTop_toTopOf="parent"/>

</android.support.constraint.ConstraintLayout>
```

具体使用(java):

```
package com.example.ly.testdrawable;

import android.graphics.Canvas;
import android.graphics.Paint;
import android.graphics.drawable.Drawable;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.text.Html;
import android.text.SpannableStringBuilder;
import android.text.Spanned;
import android.widget.TextView;
import java.util.ArrayList;
import java.util.List;
public class MainActivity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        TagTextView textView = findViewById(R.id.text);
        List<String> tags = new ArrayList<>();
        tags.add("置顶");
        tags.add("置顶1");
        tags.add("置顶2");
        textView.setContentAndTag("小阿曼是个大美女",
            tags);
    }
}
```

关于文档:

<https://blog.csdn.net/zhongwn/article/details/51489159>
https://blog.csdn.net/weixin_28717693/article/details/81811777
https://blog.csdn.net/qc_16430735/article/details/50386502
https://blog.csdn.net/qc_16430735/article/details/50427978
<https://www.cnblogs.com/tianzhijiexian/p/4297664.html>

各种span:

<https://www.jianshu.com/p/472fd3e32324>