Can I underline text in an android layout?

150

android android-layout fonts

How can I define underlined text in an Android layout xml file?

share improve this question

Janusz
53k●65●192●269

Asked
Mar 7 '10 at 2:26

Edited
Feb 2 '12 at 13:03

7 Answers

order by votes

291

It can be achieved if you are using a **string resource** xml file, which supports HTML tags like , and <u></u>.

```
<resource>
   <string name="your_string_here">This is an <u>underline</u>.</string>
</resources>
```

If you want to underline something from code use:

```
TextView textView = (TextView) view.findViewById(R.id.textview);
SpannableString content = new SpannableString("Content");
content.setSpan(new UnderlineSpan(), 0, content.length(), 0);
textView.setText(content);
```

share improve this answer

Anthony Forloney
29.8k●9●70●85

Answered
Mar 7 '10 at 2:29

Janusz
53k●65●192●269

Edited
Mar 9 '10 at 8:12

14 Hi, i tried the above resource xml code and it diddnt work. it continued to display the <u>underline</u> as a plain text – **jonney** Jul 1 '11 at 10:15

3 See also: android.text.Html.fromHtml( ) which returns a Spanned. – **Mark Renouf** Sep 7 '11 at 18:21

You should type it in strings.xml file itself, not doing it by add button. Otherwise you will get this <u> in your strings.xml file and <u>underline</u> in your code – **gdrt94** Apr 1 at 18:13

I have come across cases when underlying through <u> tags does not work, e.g. sometimes if you are using a custom font. However, underlying programmatically by UnderlineSpan has yet to fail on me, so I would recommend it as the most reliable solution. – **Giulio Piancastelli** Apr 2 at 18:17

Writing in the xml resource file did not work for me on Android SDK 4.0.3 . But Java code worked. Thank you – **Nilesh** May 16 at 13:18

add a comment

120

You can try with

```
textview.setPaintFlags(textview.getPaintFlags() |  Paint.UNDERLINE_TEXT_FLAG);
```

share improve this answer

vado
1,201 1 4 2

Answered
Jun 8 '12 at 10:44

Guykun
1,753 9 20

Edited
Jun 28 '12 at 16:00

13 This is by far the easiest solution. This should be the accepted answer. – **Nightwish1986** Feb 15 '13 at 9:36

Maybe a minor point, but OP wanted this defined in XML layout file; otherwise, this is a great solution. – **kwishnu** Jul 27 '13 at 5:43

2 use this solution if you want to clear it – **breceivemail** Aug 17 '13 at 9:19

add a comment

19

The "accepted" answer above does **NOT** work (when you try to use the string like textView.setText(Html.fromHtml(String.format(getString(...), ...))).

As stated in the documentations you must escape (html entity encoded) opening bracket of the inner tags with <, e.g. result should look like:

```
<resource>
   <string name="your_string_here">This is an &lt;u>underline&lt;/u>.</string>
</resources>
```

Then in your code you can set the text with:

```
TextView textView = (TextView) view.findViewById(R.id.textview);
textView.setText(Html.fromHtml(String.format(getString(R.id.textview), ...)));
```

share improve this answer

Ogre_BGR
4,729 1 21 30

Answered
Mar 31 '12 at 10:22

Edited
May 22 '12 at 14:10

<u>underlined here</u> works perfectly. <u>underline</u> did not work. This is for Android 2.2. Maybe different versions interpret it differently. – **autotravis** May 14 '12 at 1:13

@autotravis which one is for 2.2? I did not tested it on older versions and it will be quite unfortunate if different versions handle it differently... Also at least current documentation states that it have to be escaped (link is in the answer). – **Ogre_BGR** May 18 '12 at 8:07

I've tested <u>underlined</u> on android 2.3 and it works. <u>underline</u> does not work for 2.3. The docs say "Sometimes you may want to create a styled text resource that is also used as a format string. Normally, this won't work because the String.format(String, Object...) method will strip all the style information from the string. The work-around to this is to write the HTML tags with escaped entities, which are then recovered with fromHtml(String), after the formatting takes place." So I guess you would use the escaping if you run it through that String.format(...) method. – **autotravis** May 19 '12 at 19:07

@autotravis yes, you are correct. I use it with String.format(...). I've edited my answer, thank you for your feedback. – **Ogre_BGR** May 22 '12 at 14:08

On 2.3 and 4.1 (only ones I tried so far) you can just use textView.setText(getText(R.string.text)) instead of having to use getString(), Html.fromHtml() and String.format(). – **RoyS** Jul 24 '12 at 7:50

show **1** more comment

9

Strings.xml file content:

```
<resource>
    <string name="my_text">This is an <u>underline</u>.</string>
</resources>
```

Layout xml file shold use the above string resource with below properties of textview, as shown below:

```
<TextView android:layout_width="fill_parent"
    android:gravity="center_horizontal"
        android:layout_height="wrap_content"
    android:selectAllOnFocus="false"
        android:linksClickable="false"
    android:autoLink="all"
        android:text="@string/my_text"
/>
```

share improve this answer

Devendra Anurag
91 1 1

Answered
Apr 11 '12 at 6:04

Sahil Mahajan Mj
5,884 2 13 52

Edited
Dec 11 '12 at 9:34

1 Make sure you edit the string resource file inside the actual XML rather than inside the helper editing UI in Eclipse, at it will escape the <s. – **Chris R** Jan 23 '13 at 22:04

add a comment

2 I know this is a late answer, but I came up with a solution that works pretty well... I took the answer from Anthony Forloney for underlining text in code and created a subclass of TextView that handles that for you. Then you can just use the subclass in XML whenever you want to have an underlined TextView.

Here is the class I created:

```
import android.content.Context;
import android.text.Editable;
import android.text.SpannableString;
import android.text.TextWatcher;
import android.text.style.UnderlineSpan;
import android.util.AttributeSet;
import android.widget.TextView;

/**
 * Created with IntelliJ IDEA.
 * User: Justin
 * Date: 9/11/13
 * Time: 1:10 AM
 */
public class UnderlineTextView extends TextView
{
    private boolean m_modifyingText = false;

    public UnderlineTextView(Context context)
    {
        super(context);
        init();
    }

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

    public UnderlineTextView(Context context, AttributeSet attrs, int defStyle)
    {
        super(context, attrs, defStyle);
        init();
    }

    private void init()
    {
        addTextChangedListener(new TextWatcher()
        {
```

```
        @Override
        public void beforeTextChanged(CharSequence s, int start, int count, int after)
        {
            //Do nothing here... we don't care
        }

        @Override
        public void onTextChanged(CharSequence s, int start, int before, int count)
        {
            //Do nothing here... we don't care
        }

        @Override
        public void afterTextChanged(Editable s)
        {
            if (m_modifyingText)
                return;

            underlineText();
        }
    });

    underlineText();
}

private void underlineText()
{
    if (m_modifyingText)
        return;

    m_modifyingText = true;

    SpannableString content = new SpannableString(getText());
    content.setSpan(new UnderlineSpan(), 0, content.length(), 0);
    setText(content);

    m_modifyingText = false;
}
}
```

Now... whenever you want to create an underlined textview in XML, you just do the following:

```
<com.your.package.name.UnderlineTextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_gravity="center_horizontal"
    android:gravity="center"
    android:text="This text is underlined"
    android:textColor="@color/blue_light"
    android:textSize="12sp"
    android:textStyle="italic"/>
```

I have added additional options in this XML snippet to show that my example works with changing the text color, size, and style...

Hope this helps!

Justin
2,152 3 19 27

Answered
Sep 11 '13 at 7:41

While this example works, I quickly realized that if you ever want to have additional control in XML that this just won't do... I have switched to a better solution that involves the following steps: 1) Subclass textview 2) Add support to underline the text via custom attributes to do the underlining as specified above. The only difference is that you only execute the underline code if the custom attribute is set. – **Justin** Mar 14 at 20:18

add a comment

0

I had a problem where I'm using a custom font and the underline created with the resource file trick (<u>Underlined text</u>) did work but Android managed to transform the underline to a sort of strike trough.

I used this answer to draw a border below the textview myself: http://stackoverflow.com/a/10732993/664449. Obviously this doesn't work for partial underlined text or multilined text.

share improve this answer

Rick Pastoor
1,045 5 13

Answered
Jul 12 '13 at 9:00

0

try this code

in XML

```
<resource>
 <string name="my_text"><![CDATA[This is an <u>underline</u>]]></string>
</resources>
```

in Code

```
TextView textView = (TextView) view.findViewById(R.id.textview);
textView.setText(Html.fromHtml(getString(R.string.my_text)));
```

Good Luck!

share improve this answer

Andrew.J
1

Answered
Aug 4 at 11:11

Prag's シ
1,685 1 3 19

Edited
Aug 4 at 11:18

Your Answer

log in

or

Name

Email

Home Page

By posting your answer, you agree to the privacy policy and terms of service.

Post Your Answer