Using Spannable Strings in Android

May 28th, 2013

Users want to use beautiful and well-designed apps. Appearance of a text in TextView in android can be changed by using html page or creating spannable string. This post tells about the latter. For example, we can modify any part of the string as shown on this screenshot:

If you want to change text appearance dynamically then spans are more preferred then html pages. Spannable strings are easy to create and use. We must simply create and assign a span to a part of a string. I'll create an app demonstrating it. Initially on the screen there is a TextView with three buttons. First two buttons change text appearance by using spans and the third button clears any previously created spans.

In this app I use only two spans: ForegroundColorSpan to make red text and UnderlineSpan to underline text. And of course there are a way more spans you can use. Let me describe some of them:

- RelativeSizeSpan to change text size
- StrikethroughSpan to strike through the text
- URLSpan to make URLs in text
- ClickableSpan to make text clickable (onClick method will be called)
- StyleSpan to make text bold and/or italic
- etc.

Other spans can be known by looking at subclasses of ParcelableSpan.

ClickableSpan and URLSpan require movement method to be set:

```
textView.setMovementMethod(LinkMovementMethod.getInstance());
```

Spannable text is created by assigning spans to spannable string. So first we must create a string with text that is able to apply spans. Let's use SpannableString for it:

```
String text = "Words in this sentence can be colored in red or underlined.";
SpannableString spannable = new SpannableString(text);
```

Now when we have spannable string we can assign spans to it by calling Spannable.setSpan(Object span, int start, int end, int flags)) method. First parameter in this method is simply one of spans described earlier. Second and third parameters specify boundaries of span. The last parameter is flag to specify how text should behave when spannable string changes. It is not used for static text and we can set it to 0.

To remove previously added span call Spannable.removeSpan(Object span)) on it.

Once spannable string is ready we can assign it to TextView. You must tell TextView that string you created is spannable string:

```
textView.setText(spannableText, BufferType.SPANNABLE);
```

Spannable strings are simple and easy to use feature that helps you to make your app nicer.

As always app source code is available on GitHub.

Posted by Ivan Davletshin May 28th, 2013 android, spannable, string, text

Tweet

« Using ViewStub on Android Parcelable in PendingIntent »

Comments

About Me

My name is Ivan Davletshin. I am a software developer and I like programming. More...

Recent Posts

- Parcelable in PendingIntent
- Using spannable strings in android
- Using ViewStub on Android

- Using Quantity Strings in Android
- Marking Incorrectly Filled Fields in Forms on Android

GitHub Repos

- Status updating...

@J-rooft on GitHub

Google+