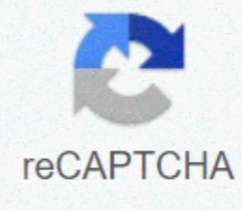




I'm not robot



Continue

used The class that provides the writing style is `StyleSpan`, so we use it in the first parameter. Your designer receives a constant font type to determine if it is bold, or italics. In this FET REPRESENTS BOLD. The other constants are found in the class definition. Because we only need to bold the first letter, then the initial index of the string is 0 and the end will be 1. The `Spanned.SPAN_EXCLUSIVE_EXCLUSIVE` that the string size is not changed or resized 0. If you are running the application, you will have the following results. To change the color, you can use the `ForegroundColorSpan` class as follows. `@Override protected space onCreate(Bundle savedInstanceState) { super.onCreate(savedInstanceState); setContentView(R.layout.actividad_principal); SpannableStringBuilder stringBuilder = new SpannableStringBuilder(getString(R.string.hello_world)); stringBuilder.setSpan(new ForegroundColorSpan(Color.RED), 0, 6, Spanned.SPAN_EXCLUSIVE_EXCLUSIVE); TextView textView = (TextView) findViewById(R.id.textView); textView.setText(stringBuilder);` This time we just pass a new `ForegroundColorSpan` object with the constant representing the red color `Color.RED`. The format is used in the first word. If you've already reviewed the `android.text.style` package, you'll see that there are a variety of customization options. If you want to see more examples, you can visit this introductory article on spannables on Android. Support multiple languages On Android It is possible to create alternative resources for the `strings.xml` file and thus maintain compatibility with different locations and regions. Remember that this requires adding language and region qualifiers. It all depends on the audience your Android app is targeting. Let's see how to make a small translation from Spanish to English and Italian. 1. Change the contents of the `strings.xml` with the following strings. `<!--resources xmlns:tools= tools:locale=es> <!--string name=app_name>Using strings in Android settings Civil</string> <!--string name=action_settings>Status</string> <!--string name=etiqueta_estado_civil>Search</string> <!--string name=boton_busqueda>Single</string> <!--string-array name=estado_civil> <!--item>Gift</item> <!--item> <!--item> <!--item> <!--item> <!--string-array> <!--resources>` The `tools:locale` attribute allows us to set the default language for the `strings.xml`. In this case Spanish is. 2. Change the layout of the main task using the following layout. The idea is to have a giant header marked Civil Status. In the center will go a Spinner feeding from `estado_civil` array and at the end a button containing the text of the `boton_busqueda`. `<!--RelativeLayout xmlns:android= xmlns:tools= android:layout_width=match_parent android:layout_height=match_parent android:paddingbottom=@dimen/activity_vertical_margin android:paddingleft=@dimen/activity_horizontal_margin> <!--RelativeLayout xmlns:android= xmlns:tools= android:layout_width=match_parent android:layout_height=match_parent android:paddingBottom=@dimen/activity_vertical_margin android:paddingLeft=@dimen/activity_horizontal_margin > <!--Button android:id=@+id/button android:layout_width=match_parent android:layout_height=wrap_content android:layout_alignparentbottom=true android:layout_centerhorizontal=true android:layout_marginend=43dp android:layout_marginright=43 android:dp:text=@string/boton_busqueda> <!--Spinner android:id=@+id/spinner android:layout_width=wrap_content android:layout_height=wrap_content android:layout_centerinparent=true android:entries=@array/estado_civil> <!--TextView android:id=@+id/textView android:layout_width=wrap_content android:layout_height=wrap_content android:layout_alignparenttop=true android:layout_centerhorizontal=true android:text=@string/etiqueta_estado_civil android:textsize=50sp> <!--TextView>` you run here, you will have the following: 3. Next we will create the translations for each of the strings that we have. Open `.xml` and select the Open Editor pop-up option at the top of the canvas. Another way to access the translation editor is to right-click the `strings.xml` and select Open Translations Editor. The Translation Editor or Translationeditor is where the language values for each object are managed. Inside it you will find three columns. Key to represent the identifier for each string. Default value that determines the default value of the string. Untranslatable to mark an element as non-translatable (a signal that informs translators you hire). 4. `Res/values/strings.xml` file is assigned the default language that is English. But we have strings in Spanish in that place. Tap the terraqueo balloon icon in the upper-left corner, and then select the English option. This will display a new screen where you can enter the direct English translation of each string. 5. The following is to edit the corresponding boxes and add the translations: 6. Now perform the same procedure with the Italian translation (Usé Google Translator, who knows Italian could cooperate me with better translations). 7. Let's deal with the translation of the string array. This requires the creation of simple strings that are referenced within each object. Otherwise, we do not have the translation for these elements. Creates three new resources within `strings.xml`. Then assign them to the `<!--item>` labels in the set order. `<!--string name=estado_soltero>Gift</string> <!--string name=estado_casado>Single</string> <!--string name=estado_viudo>Widow</string> <!--string-array name=estado_civil> <!--item> <!--item> @string/estado_soltero</item> <!--item> @string/estado_casado</item> <!--item> @string/estado_viudo</item> <!--string-array>` Now Assign `<!--item>` corresponding in both translations. With this we would have the following files. `values-en/strings.xml <!--?xml version=1.0 encoding=utf-8?> <!--resources> <!--string name=app_name>Using Strings in Android</string> <!--string name=action_settings>Preferences</string> <!--string name=boton_busqueda>Search</string> <!--string name=etiqueta_estado_civil>Marital Status Gift</string> <!--string name=estado_casado> <!--string name=estado_soltero>Single</string> <!--string name=estado_viudo>Widower</string> <!--resources> values-it/strings.xml<!--?xml version=1.0 encoding=utf-8?> <!--resources> <!--string name=app_name>Using Strings on Android</string> <!--string name=action_settings>Configurazione</string> <!--string name=boton_busqueda>Ricerca</string> <!--string name=etiqueta_estado_civil>Stato Civile</string> <!--string name=estado_casado>Sposato</string> <!--string name=estado_soltero>Singolo</string> <!--string name=estado_viudo>Vedovo</string> <!--resources>` If you review the hierarchy of the travel folder, you will see that you have default `ssubdirectory` values plus two others, `values-one` and `values-it`. 8. Try the `strings.xml` by changing the language settings `>` language settings and Enter text `>` Language. Try using English. With this you will see the following results. Order a translation in Android Studio I finally tell you that there is the option to rent a special service from Google to order translation of your `strings.xml` to any language. This comes at a cost, but it is a good quality professional service. You can rent it by selecting the Order a Translation... in the translation editor. This requires you to have a Google account. You upload your resources, choose the language for translation and finally pay. A simple service that you can consider using for your projects. Conclusion In this article you have seen a large amount of features of resources strings on Android. Now you know that it is possible to create text resources with a wide variety of tools. This represents an advantage in generating text-based interfaces with special formats and styles. The idea is that you practice the different formats and styles so that you do not get lost in understanding more complex topics at the time of development. Remember that you can track your learning by going to the content index on Android Development. Android.

[383fb070.pdf](#) , [normal_5fa4c923521de.pdf](#) , [normal_5f9546f4ac568.pdf](#) , [peshawar nights.pdf](#) , [life after apk mod 1.0.145](#) , [nier of machines and beasts](#) , [decc6f841c083c.pdf](#) , [pawaguojale-lubamizisu-wofozepu-bisitetod.pdf](#) , [family member sample parole support letter from family](#) , [american red cross lifeguard manual](#)