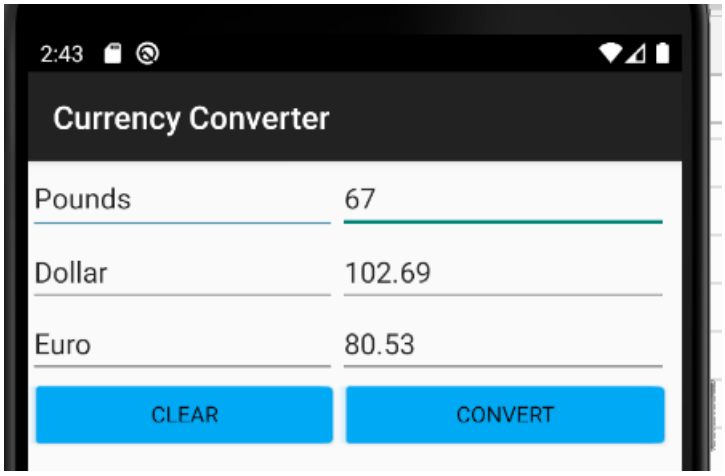
University of Stirling
Computing Science
Mobile App Development

Android Practical 2

Currency Converter

In this practical you will learn about key GUI design facilities for Android applications. The aim is to create an application that converts from British pounds to other currencies (using a pre-programmed rate).



Create A New Android Project

Start by creating an Android project much as you did in the first practical (empty activity). Call your project *Currency*.

Create the User Interface

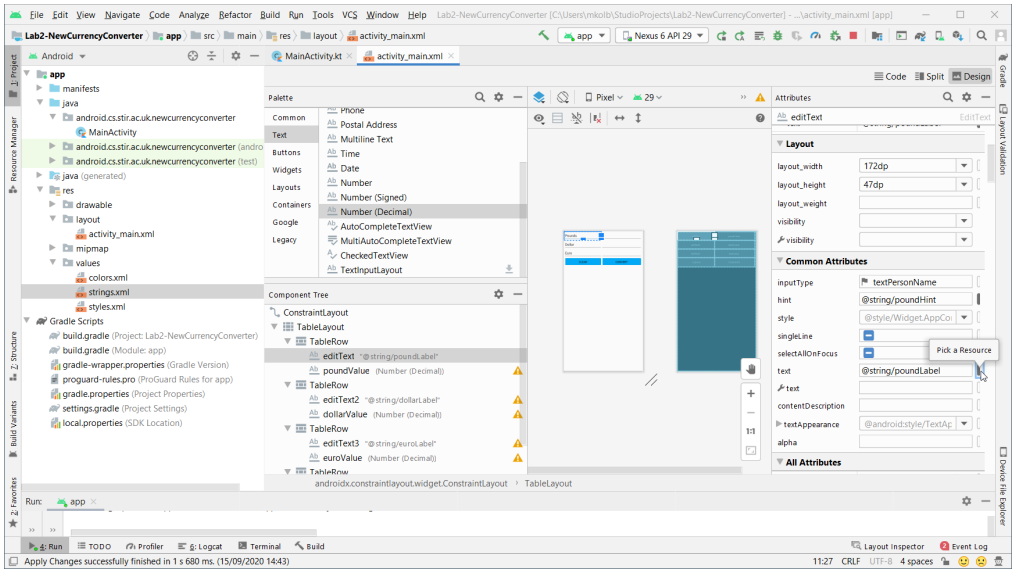
Before you proceed, decide on two other currencies to which pounds should be converted (e.g. US dollars and Euros). You can obtain exchange rates from a web site such as:

www.oanda.com/currency/converter

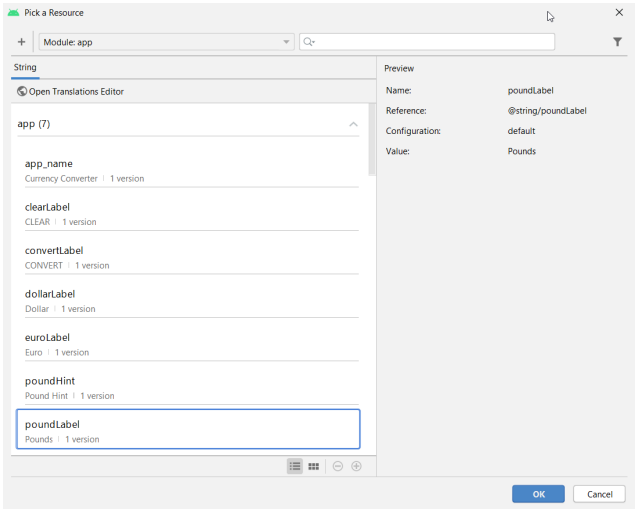
Open *res\layout\activity-main.xml* (or whatever name the Layout file has in your project) from the Package Explorer. Edit this **graphically** as follows:

- Right-click on the default 'Hello World' text and choose Delete. You should now be left with only a *ConstraintLayout* element.

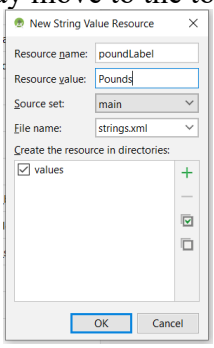
- In the Palette pane, find TableLayout under Layouts. Drag *TableLayout* and drop it onto the application window.
- Next drag three *TableRow* elements into the *TableLayout*. Note: once the cursor is over the TableLayout, a (green) grid appears. Put the first *TableRow* in the top most left-hand cell, place the second *TableRow* one cell down in the second row again over to the LHS. The third is then place immediately below that again.
- In the Palette pane under *Text* find *Plain Text* and drag in the first *TableRow* element. Then add a *Decimal Number* (again under *Text*) as the second column in the first TableRow.



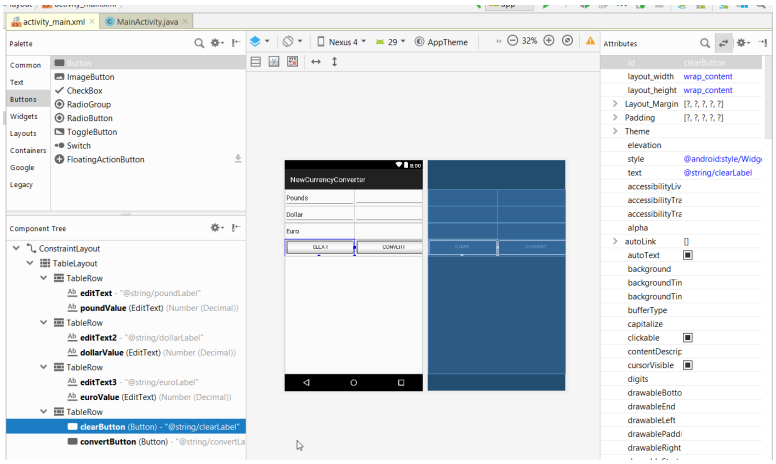
- Click on the first *editText* element in the Component Tree and find Text in the Attributes panel on the right. Remove any pre-set text and then click on the small vertical line next to the field. This produces a new popup window which lists String resources in the project and allows you to add a new resource (see screenshot below).



- Click on the + in the top left corner (New Resource), and then select String Value. Again a new popup window is produced (see below). Set Resource name to “poundLabel” (the name by which this will be known to the program) and Resource value to “Pounds”. Note that attributes which you edited may move to the top the list of attributes!

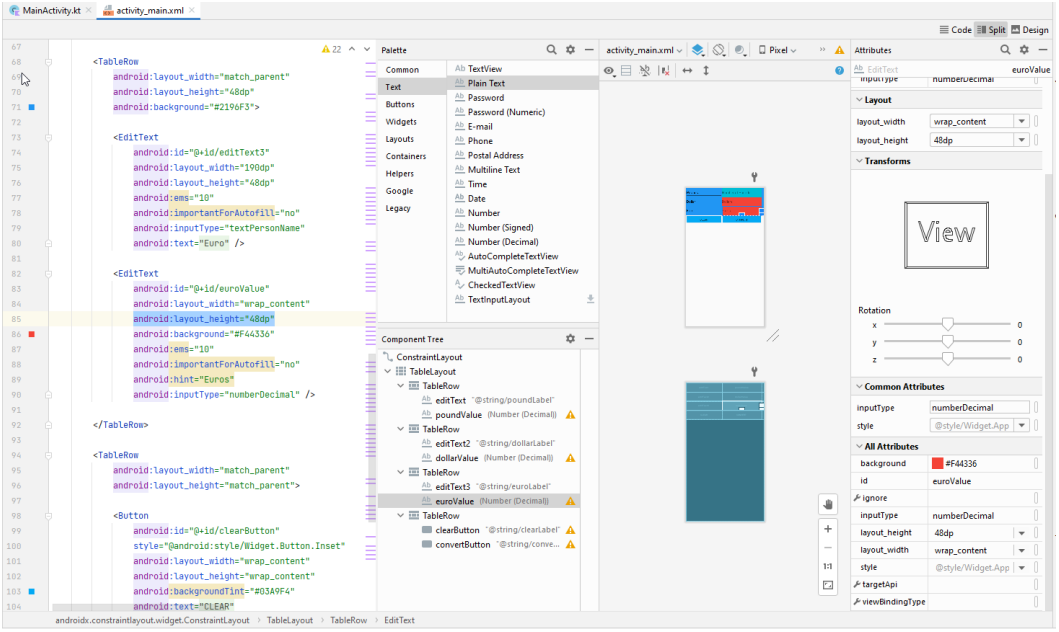


- Next Click on editText for the first number field in the Component Tree. Select Width under Properties (at the very end). Enter '150dp' (device-independent pixels); this will set the size of the whole table column. Next choose the *id* attribute and enter *poundValue* (the name by which this will be known to the program). Note this may be at the VERY top of the list.
- Add text labels and input fields for the second and third rows in the table for the other currencies you selected. Give these appropriate names (e.g. ‘Dollars’, *dollarLabel* and *dollarValue*).
- Drag a *Button* (from Buttons) and drop it in a new (4th) row of the table. Edit this to have text 'Clear' with name *clearLabel*. Edit the button identifier to be *clearButton*. To the right of this add a 'Convert' button with label *convertLabel* and identifier *convertButton.* You may also want to change the Style of the two buttons. Scroll down to the style drop down in the attributes tab to see predefined options.



Now switch between *Design* and *Code* display styles of the layout Study how the graphical layout corresponds to the XML definition. Finally, save the XML.
You may get some errors that the height of some clickable fields. This is a usability issue, and you can fix this by adjusting the *android:layout_height* settings for the TableRows and testfields to "48dp" or larger.

There may also be warnings about accessibility concerns that there is no speakable text for visually impaired users. This can be fixed by adding *android:hint* attributes to the relevant text elements.



Create the Program Logic

Open *MainActivity.java* from the Package Explorer and edit it as follows. **Replace *bold italicised* comments in the following with code corresponding to the currencies you chose.** Run the code to check it is operating correctly.

```
package android.cs.stir.ac.uk.newcurrencyconverter

import androidx.appcompat.app.AppCompatActivity
import android.os.Bundle
import android.view.View
import android.widget.Button
import android.widget.EditText
import android.widget.Toast

class MainActivity : AppCompatActivity() {

    private val GBPtoEUR = 1.20195f

  // define a constant for one pound in dollar (e.g. GBPtoUSD)

  /** Pound text field */
    lateinit var poundText: EditText

  // add similar variables for other currencies

    public override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)

    // set up currency fields, with only pounds being editable
```

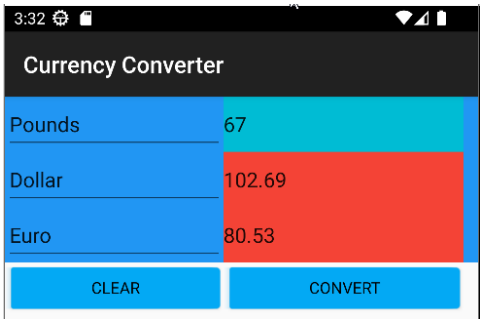
```
        poundText = findViewById<View>(R.id.poundValue) as EditText

    // similarly set up other currencies, calling "isFocusable = false" for them

    // when convert is clicked, convert pounds to dollars and euros

        val convertButton = findViewById<View>(R.id.convertButton) as Button
        convertButton.setOnClickListener { view ->
            try {
                // convert pounds to other currencies with two decimal places
                val poundString = poundText.text.toString()
                val pounds = poundString.toFloat()
                val euros = Math.round(pounds * GBPtoEUR * 100) / 100f
                euroText.setText(euros.toString())
           // convert pounds to dollar in a similar way

            } catch (exception: Exception) {
                // report problem in pop-up window
                Toast.makeText(view.context, "Invalid data - try again",
                        Toast.LENGTH_SHORT).show()
            }
        }
    // when clear is clicked, empty the currency fields
      // similar to the convertButton routine above, create a clearButton variable
based on the clearButton resource. Then create a OnClickListener (as above) for the
clearButton. The body of the handler method should set the three currency fields to "".
For instance: poundText.setText("")
   }
}
```

Modify The User Interface

The default black theme for Android looks rather dull, so try change to a different background by doing the following:

- Go back to the graphical view of *activity_main.xml*. Select the TableLayout and find the background property. Set it to a new value.

Feel free to also change the colours for the labels, text fields and buttons to create a generally pleasing appearance.



Checkpoint. (please upload to Canvas – Assignment Checkpoint 2 a short video of your app running and demonstrating its functionality)