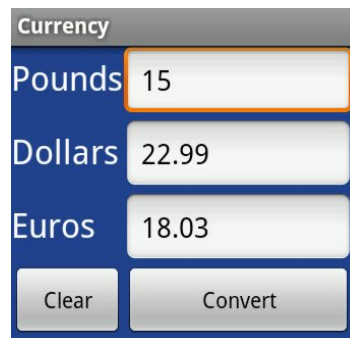


University of Stirling Computing Science Mobile App Development

Android Practical 2

Currency Converter

In this practical you will learn about GUI design 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. 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:

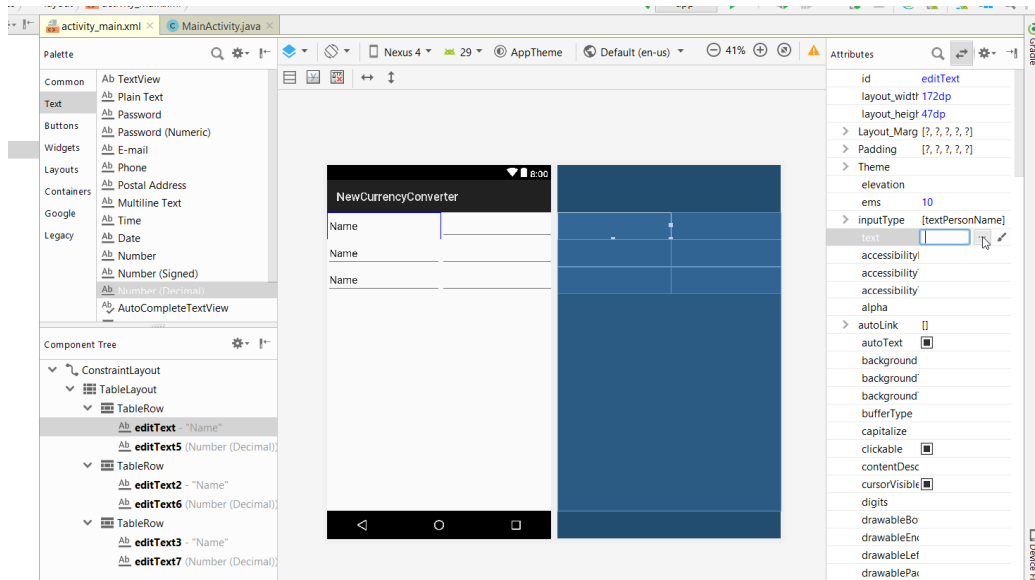
www.oanda.com/currency/converter

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:

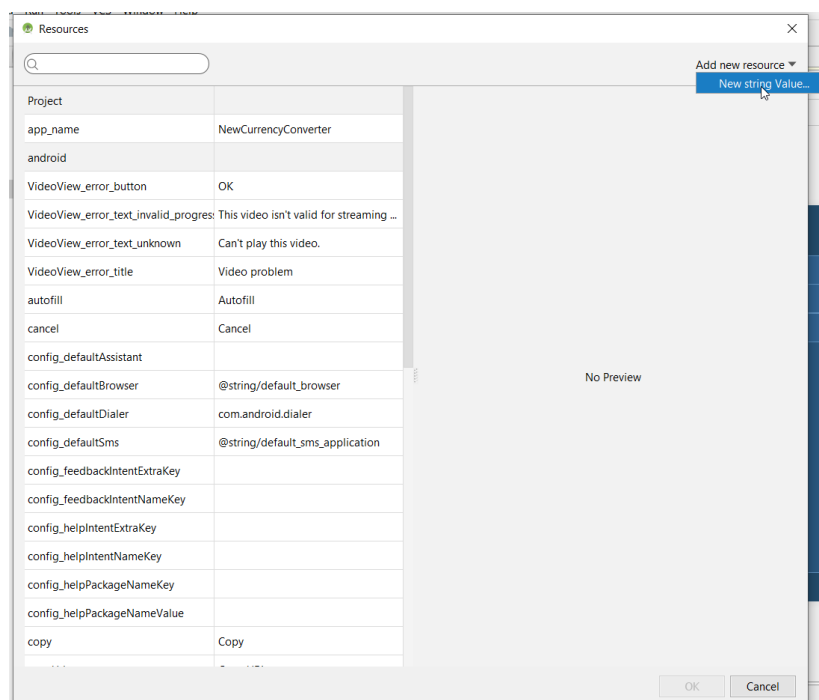
- Right-click on the default 'Hello World' text and choose Delete. You should now be left with only a *ConstraintLayout* element.
- In the Palette pane, find *TableLayout* under Layouts. Drag *TableLayout* and drop it onto the application window.
- Next drag three *TableRow* elements into the *TableLayout*. 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.

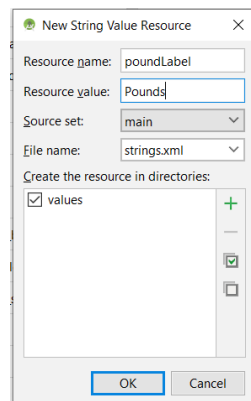
- In the Palette pane under *Text* find *Plain Text* and drag in the first *TableRow* element. Then add a *Decimal Number* (again under *Text*) as the second column in the first *TableRow*.



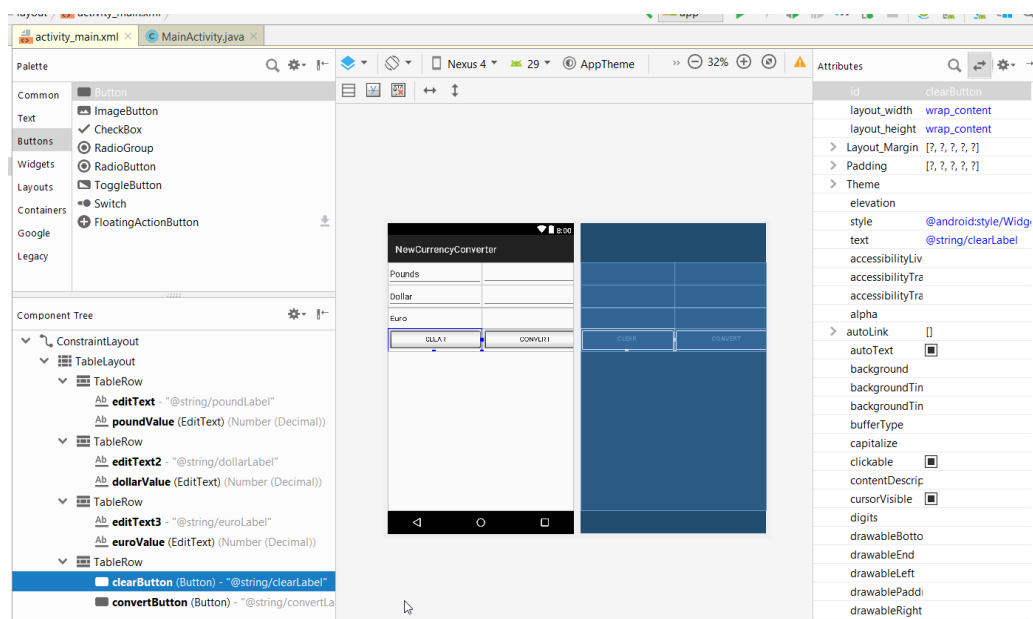
- Click on the first *editText* element in the Component Tree and find *Text* in the Attributes panel on the right. Remove any pre-set text and then click on the three dots. This produces a new popup window which lists String resources in the project and allows you to add a new resource (see screenshot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



- Click on New Resource. Again a new popup window is produced (see above). 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. Click on the little pen symbol in the Attributes panel to see predefined options.



Now switch between *Graphical Layout* and *Text*. Study how the graphical layout corresponds to the XML definition. Finally, save the XML.

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 uk.ac.stir.cs.android;

import android.app.Activity;           // import activity
import android.os.Bundle;             // import bundle
import android.view.View;             // import view
import android.widget.Button;          // import button
import android.widget.EditText;        // import edit text
import android.widget.Toast;          // import toast

public class MainActivity extends Activity {

    /* ----- Constants ----- */

    // define constants for one pound in other currencies (e.g. GBPTtoXYZ)

    /* ----- Variables ----- */

    /** Pound text field */
    EditText poundText;

    // add similar fields for other currencies

    /* ----- Methods ----- */

    /**
     * Create user interface and set up listeners for buttons.
     *
     * @param savedInstanceState    previously saved state
     */
    @Override
    public void onCreate(Bundle savedInstanceState) {
        // create basic interface
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        // set up currency fields, with only pounds being focusable (i.e. editable)
        poundText = (EditText) findViewById(R.id.poundValue);
        // similarly set up other currencies, calling "setFocusable(false)" for them

        // when convert is clicked, convert pounds to other currencies
        Button convertButton = (Button) findViewById(R.id.convertButton);
        convertButton.setOnClickListener(new OnClickListener() {
            public void onClick(View view) {
                try {
                    // convert pounds to other currencies with two decimal places
                    String poundString = poundText.getText().toString();
                    float pounds = Float.parseFloat(poundString);
                    // convert pounds to other currencies as follows
                    // float xyz = Math.round(pounds * GBPTtoXYZ * 100) / 100f;
                    // xyzText.setText(String.valueOf(xyz));
                }
                catch (Exception exception) {

```

```

        // report problem in pop-up window
        Toast.makeText(view.getContext(), "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 (see screenshot on the left), 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. Click on the ... and define a new resource in the window which opens up (make sure you are on the Project panel). Name it *backgroundColour* and value *#224488*.

Feel free to also change the colours for the labels, text fields and buttons to create a generally pleasing appearance. Can you achieve the look from the first screenshot on this lab sheet (page 1)?

Checkpoint. Answer the following questions:

- Is it preferable to edit the layout graphically or in XML format?
- Why are currencies except pounds set as non-focusable?
- Consider (but don't code) how you would avoid having pre-defined exchange rates.