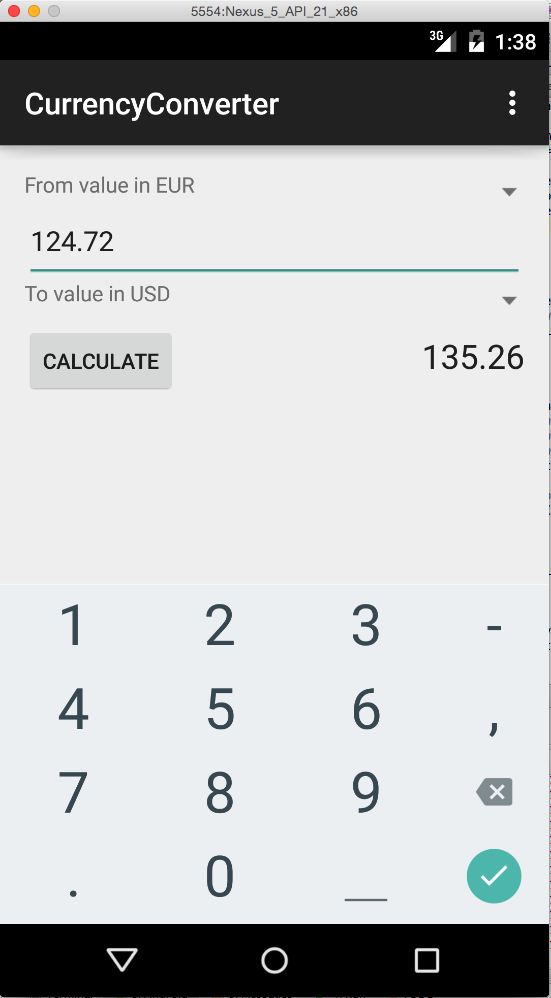
## Exercise 2.1: Currency Converter (Basic Layout)

You will create a small own app to convert amounts of money from one curency (f.e. Euro) to another currency (f.e. US dollar).

The user interface might look like shown in the picture on the right. The layout consists of the following components:

ConstraintLayout as base layout of the *Activity* with the following rows (see example on the right):

* Text „From value in “ left of Spinner with selection of source currency
* EditText (inputType=“numberDecimal“) to input amount
* Text „To value in” left of Spinner with selection of target currency
* Button „Calculate“ triggering the conversion on the left, result of conversion to two decimal points, initially „0.00“ on the right

Create a new project with one *Activity* and design its layout!

Change between the Design- and XML-view and look up the documentation[[1]](#footnote-1) when unsure. Especially take care about the layout properties of ConstraintLayout.

For this exercise the layout should only be shown. The selection boxes (*Spinners*) for source and target currency may stay empty. The “Calculate” button is inoperable.

Feel free to experiment with the design or create something looking completely different!

## Exercise 2.2: Currency Converter (List of Currencies)

You will find two classes as a data model in the exercise materials in Moodle:

* ExchangeRate: Saves abbreviated name and exchange rate for a currency
* ExchangeRateDatabase: Manages a list of exchange rates for a set of currencies[[2]](#footnote-2). The currencies are currently stored directly with the source code. With its methods the list of currencies can be queried and an amount can be converted from one currency to another.

Integrate the classes with your project and get acquainted with them.

Use the class ExchangeRateDatabase to fill the Spinners with options:

* Create an appropriate ArrayAdapter[[3]](#footnote-3)
* Set it for the Spinner-Views (see lecture)

Your app should now be able to select two currencies and enter a decimal number

## Exercise 2.3: Currency Converter (Functionality)

Register an onClick handler with the button. On selection source- and target currency and the source amount should be used to perform a conversion and show the result to two decimal points.

Remarks:

* Single layout views can be found using findViewById(R.id.id\_of\_view).
* The selected item of a spinner can be obtained with getSelectedItem().

## Exercise 2.4: Currency List (Additional Activity)

Create an additional *Activity* with a list of all currencies (example on the right).

Create a new empty *Activity* for your project (File 🡪 New... 🡪 Activity 🡪 Empty Activity):

* Enter a good name for the *Activity* (f.e. *„CurrencyListActivity“)*
* Select the option *„Launcher Activity“*

Define a fitting layout for the ListView and its entries.

Create an ArrayAdapter that supplies the view with its entries.

When running the app the launcher might show two icons, depending on what you selected when creating the *Activity*. However, each icon will start the original *Activity*.

Try to edit *AndroidManifest.xml* such that the main *Activity* is deactivated and you can instead see the currency list. We will fix this problem in an upcoming exercise.

## Exercise 2.5: Currency List (Custom Adapter)

Extend your list and the *Spinners* such that you directly show the exchange rate in the list. For this you need to write your own adapter class as shown in the lecture.

1. f.e. <https://developer.android.com/guide/topics/ui/declaring-layout.html> and

   <https://developer.android.com/guide/topics/ui/controls/spinner.html> [↑](#footnote-ref-1)
2. Source ECB: <https://www.ecb.europa.eu/stats/exchange/eurofxref/html/index.en.html> [↑](#footnote-ref-2)
3. <https://developer.android.com/reference/android/widget/ArrayAdapter.html> [↑](#footnote-ref-3)