## Exercise 7.1: Fetch/Add data in background

Change API calls to occur in the background!

Take the following steps:

* Remove the lines that start with StrictMode... from the onCreate()-method of your *Activities*.
* Create a classes for each API call that implement the interface Runnable.
* Move the method updateCurrencies() to this class. If needed create a constructor that can take references to required objects (f.e. activity).
* Implement the run()-method of your *Runnable*.
* When the users selects the menu entry to refresh currency rates, start this process in the background by instantiating ExchangeRateUpdateRunnable and starting a new *Thread*.
* Ask yourself:
  + Which methods (in ExchangeRateDatabase) have to be synchronized?
  + What would happen, when the user starts a refresh while one is still in progress?

When in doubt, ask the advisor!

Use the Log class to see when the update is started and finished!

Do you notice that the UI stays responsive during the update now?

**Note:** You can simulate a “bad” network connection in the emulator by opening the “Extended controls” (button “…” in the toolbar) and then in the area “Cellular” selecting a worse network type or signal strength.

## Exercise 7.2: A toast to the new currency rates! (Currency Converter)



Create a toast that informs the user when the update is finished.

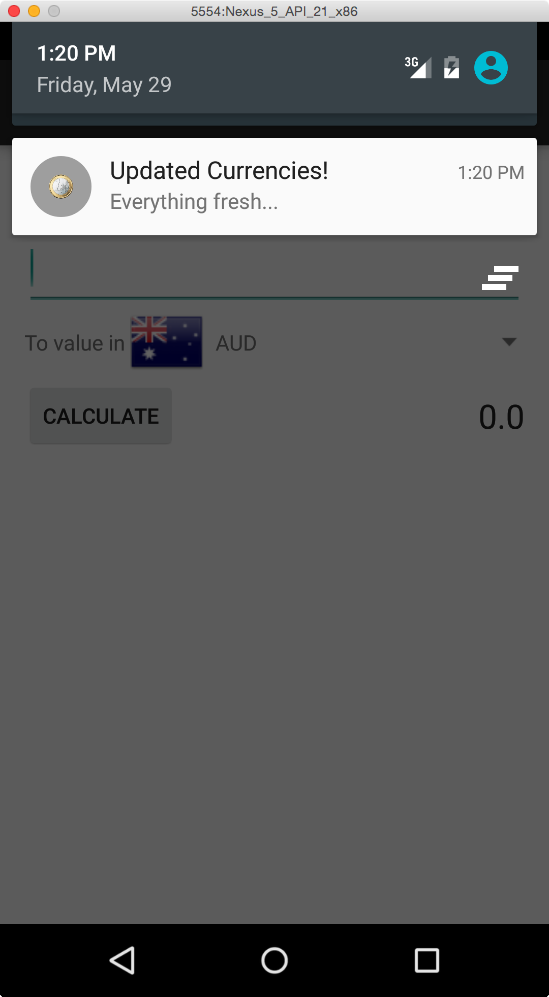
The toast is triggered from ExchangeRateUpdateRunnable.run() once the currency data has been refreshed.

However, a toast can only be started from the UI thread. This means you will need to create a new Runnable that is passed to the UI thread for execution:

**activity**.runOnUiThread(**new** Runnable() {  
 @Override  
 **public void** run() {  
 // Create and show toast!

}  
});

## Exercise 7.3: A Notification for new currency rates (Currency Converter, extra)



Show an additional *Notification* once the refresh has finished!

## Exercise 7.4: Find out how to deploy your app (Currency Converter)

Create a signed APK as shown during the lecture.