

CMSC 436 Lab 12

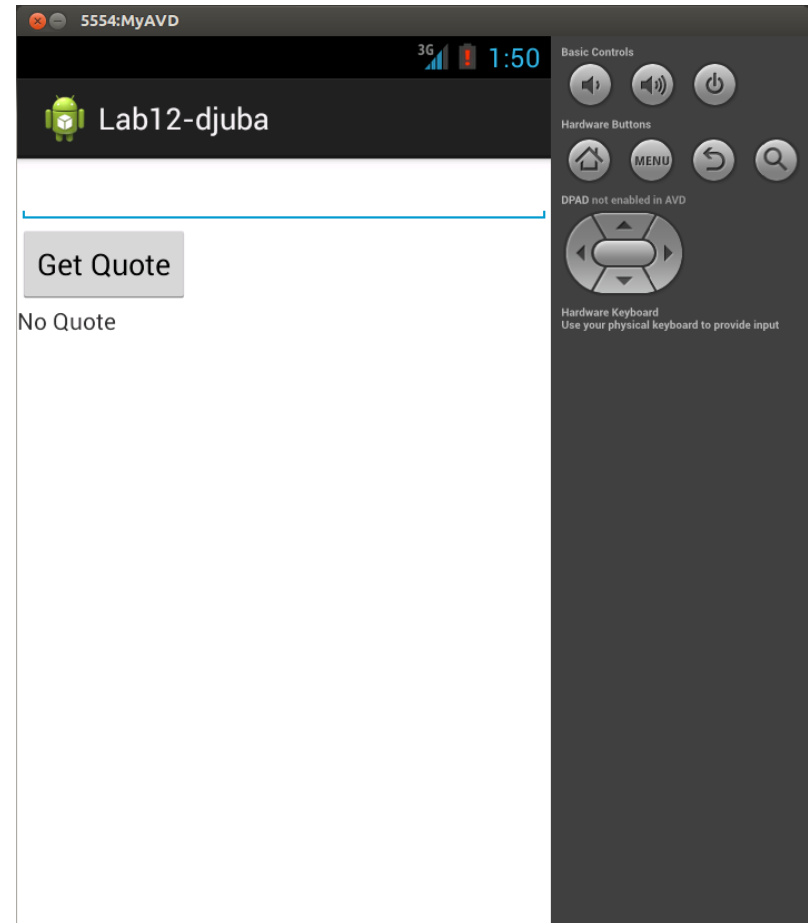
Performing Network Operations

Overview

- For this lab you will use Android networking classes to send a request to a web API over HTTP and retrieve stock quote data
- The information needed to do this lab can be found on the Android developer site at <http://developer.android.com/training/basics/network-ops/connecting.html>

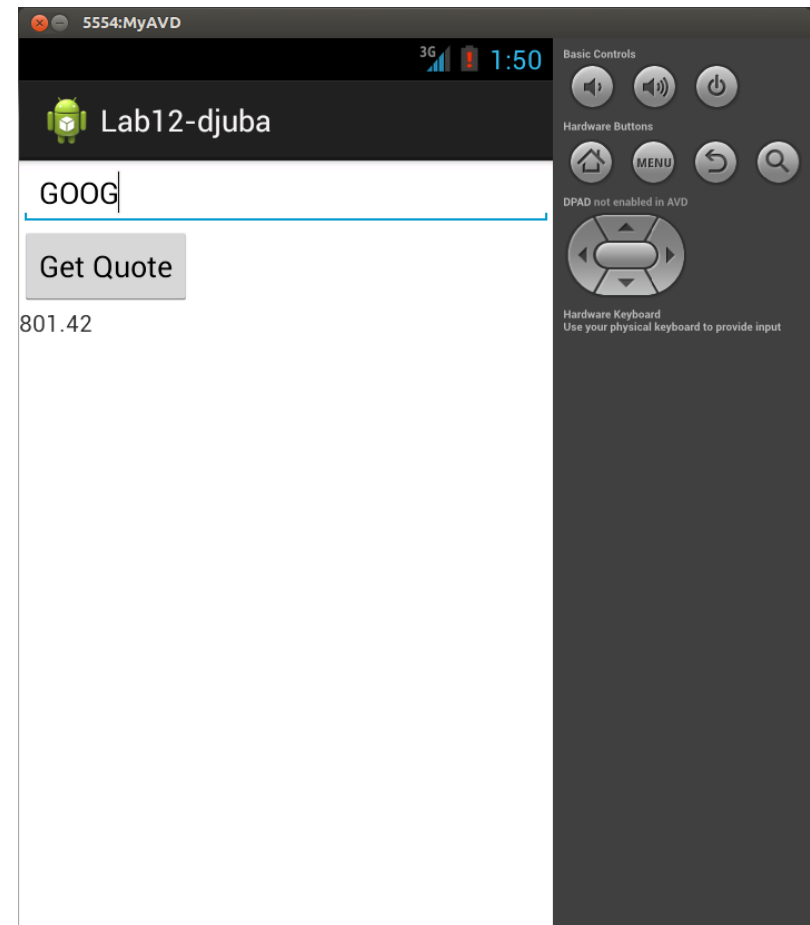
Interface

- The main activity should contain an EditText, a “Get Quote” Button, and a TextView which initially displays “No Quote”



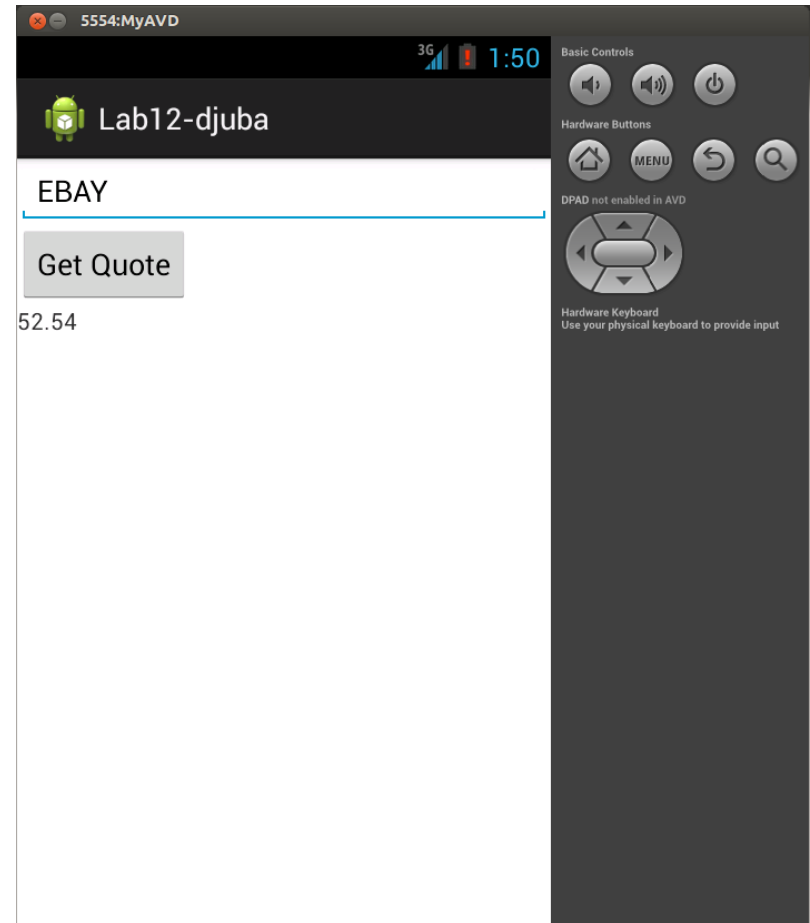
Interface

- Entering a stock symbol in the EditText and clicking the button should cause the current stock price to be displayed
- The price should automatically update once every second



Interface

- If a new symbol is entered and the button is clicked again, the price for the new stock should be displayed and updated instead



Web APIs

- Many different sites make their data available in an easy-to-parse format over the web
- These include
 - Rotten Tomatoes
<http://developer.rottentomatoes.com/>
 - eBay
<http://developer.ebay.com/common/api/>
 - Yahoo Finance
<http://developer.yahoo.com/finance/company.html>

Web APIs

- This data is often in XML format
- Android provides an implementation of the XmlPullParser interface to aid in parsing this
- For this lab, however, we will parse data from a much simpler format

Web APIs

- Yahoo Finance has an unofficial API which returns data in CSV (Comma Separated Value) format, documented here
<http://www.gummy-stuff.org/Yahoo-data.htm>
- If you request only a single value (such as a stock price), the result returned is only a single number, which is trivial to parse
- Many different types of data can be requested. The URL to request the last trade price for a stock is
<http://finance.yahoo.com/d/quotes.csv?f=l1&s=SYMBOL>

Web APIs

- Note that the stock market is closed on evenings and weekends. Requesting a stock quote during these times will result in the last trade price before the market closed. Keep this in mind when testing your app.

HTTP Client

- Android provides two HTTP clients- the heavier weight Apache HttpClient and the lighter weight HttpURLConnection
- Originally HttpURLConnection was buggy and not recommended, but these issues have been resolved and now it is the only HTTP client in active development on Android
- For this lab you should use HttpURLConnection

Permissions

- Be sure to use the permissions `android.permission.INTERNET` and `android.permission.ACCESS_NETWORK_STATE`

Threading

- Network access should be done on a separate thread to avoid blocking the UI. In fact, since Honeycomb, doing network access on the main thread will cause an exception.
- You should do your threading using an AsyncTask, which you previously used in lab 7
- You can update the stock price using AsyncTask's onProgressUpdate callback

Threading

- When the Get Quote button is clicked, you should cancel any currently running AsyncTask using its `cancel()` method
- For this to work, your loop in `doInBackground()` will have to stop when `isCancelled()` becomes true
- You should also cancel the AsyncTask when the activity is destroyed, otherwise it will continue to run
- Be sure to have the thread sleep for one second between updates