

Project

The interview for Vaporstream included a Coding Challenge as outlined in the document “CODING_INSTRUCTIONS”.

Thoughts

Challenges

The main challenge was just learning the particulars of how to work with the worldweatheronline API. In general I would say they provided a fairly straight forward and easy to work with interface with minimal bugs. (There was one curiosity that I think was a bug).

One Activity not Two

The instructions implied strongly to have a search activity and then display the result in a separate activity. However, I took the liberty to instead display the results within the same page as the search activity because it seemed more natural from a UI perspective to do it this way.

Number of Entries

The instructions indicated 5 days of results. I decided to do this with 6 hour intervals for each day. And I also included on top the current conditions. Thus there are not 5 entries, but 21.

Extra Code

At first I mistakenly queries using their SEARCH API, instead of the LOCAL WEATHER api. Once I discovered my mistake I easily flipped over to the proper LOCAL WEATHER api. However, in the code you will see I retained the work of accessing the SEARCH api. I did this because I thought it would be nice to allow the app to be extended to query from all the different API's available. Not just from the LOCAL weather api. If code brevity is a concern this unnecessary code could easily be removed.

The other thing I left as comments the pulling in of a lot more available information that the site is providing even though the coding instructions only indicated that you wanted to display comparatively very little of it. I did this where it was easy to do. However, this means there is quite a bit of room for the code to be streamlined if brevity of code comments is more of a concern than indicating via comments possible enhancements in the future.

Inefficiency

The loading of the weather icon occurs directly against a download from the weather service. This is seen by the user by the icon always seeming to have to “load”. It takes less than a second, however it is noticeable. An improvement would be to buy one of the icon sets available from the weather site and

use these instead. There is a “weather code” passed back as part of the query that makes doing this easy.

Memory Issues

I had some problems with Picasso running out of memory and sometimes crashing even though I was using it right. I fixed them by using my own Bitmap cache. See BitmapHelper.java.

CODING FEATURES

3rd Party Libraries

- Butterknife: provides very simple to use abbreviations to the common Android findViewById pattern.
- EventBus (by Greenrobot): A very simple to use event bus scheme. Using this simplifies message passing between distant components. Better in certain instances than trying to establish overly cumbersome a network of listeners. I really like this library.
- Timber: A nice library to handle debug message reporting rather than Log.
- Picasso: A nice way to grab image/icons from local or remote locations.

Time

It took me just under 11 hours to complete this project.