Evan O'Brien

Synchronoss Assignment

Irish Rail demo app

Design:

- The implementation is fragment based as opposed to solely activity based for modularity (and easier portability to tablets)
- The implementation uses the **MVP design pattern** so that business logic can be separated from the view. Note that this separation is not perfect at the moment as there is some string formatting in the TrainListAdapter class.Usually I would pass a formatted object to the adapter and avoid any work in the getView function
- The implementation is very straightforward. There is currently one fragment that contains an AutocompleteTextView and a button in a section at the top of the layoiut and listview below that The AutocompleteTextView is populated with suggestions retrieved using the getAllStationsXML call[1] (This data is retrieved when the application starts)
- Clicking on the button calls back on the presenter and populates the listview with data from the getStationDataByNameXML call[2]

Libraries used:

- **Volley** [3]was used for REST queries as I have found that to be a useful approach in the past (the fact that you can pass your own http stack implementation is useful, particularly if you want to use pinning). This reduces the amount of boilerplate code and the library is very well exercised in production applications.
- **SimpleXml** [4] was used to serialize and deserialize the XML data. The class XmlRequest implements the marshalling from xml to pojos. The model classes are marked up with the relevant annotations to deal with this marshalling. This approach was chosen for it's fluency, code readability and maintenanc. A handcoded Sax implementation could have been used here but the data is not large so there would not have been of much benefit.

---

[1] http://api.irishrail.ie/realtime/
[2] api.irishrail.ie/realtime/realtime.asmx/getStationDataByNameXML?StationDesc=Bayside
[3] http://developer.android.com/training/volley/index.html
[4] http://simple.sourceforge.net/

- **ButterKnife**[5] was used for view bindings - this simply eliminates boilerplate code and makes our implementation cleaner (I have used an older version that I use in my production applications.

Unfinished Items:

- Need to add a filter (spinner) to filter underneath the AutocompleteTextView to filter the items by direction(this is a field retrieved using .
- Clean up styling
- Pull to Refresh (I have disabled this for the time being)
- Refactor the data request classes (there is duplication there)
- Save Instance state
- Improve the UI and styling.

[5] http://jakewharton.github.io/butterknife/