GitHub Username: trackdave

Currency Exchange Rate

Description

This app retrieves the foreign exchange rate for multiple countries on a daily basis as well as historic exchange rates since 1999.

Intended User

Anyone in finance or international travelers

Features

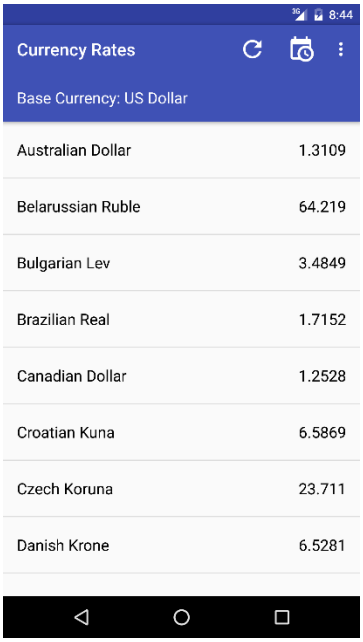
List the main features of your app. For example:

- Retrieve's currency data daily
- Retrieve's historic currency data
- Create a chart to map out currency data
- Pop up notifications for updated currency data

User Interface Mocks

These can be created by hand (take a photo of your drawings and insert them in this flow), or using a program like Photoshop or Balsamiq.

Screen 1

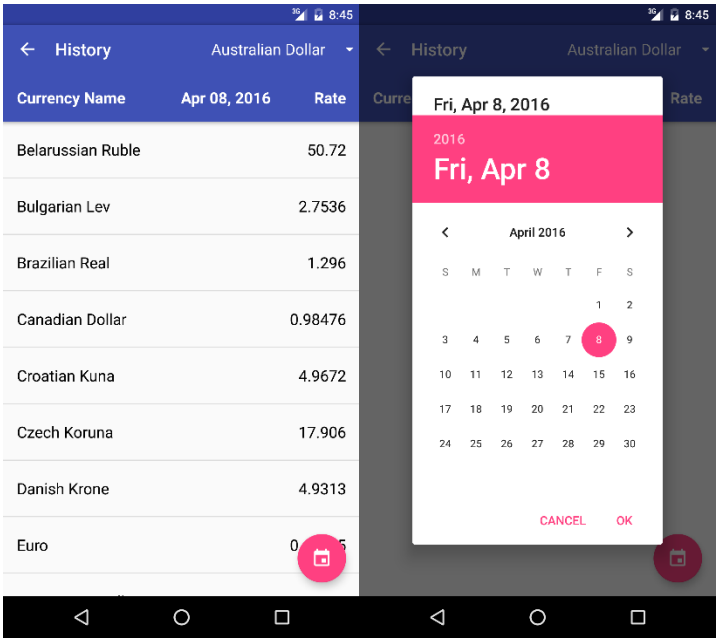


This is the main activity. The main activity will contain a list of target currency rates vs the base currency rate selected by the user in the settings screen (default US Dollar). This has menu options to refresh the data, go to the history activity, and go to the settings activity. Clicking an item on the list will load the chart activity to view a chart of the base currency vs the target currency (clicked item).

Screen 2

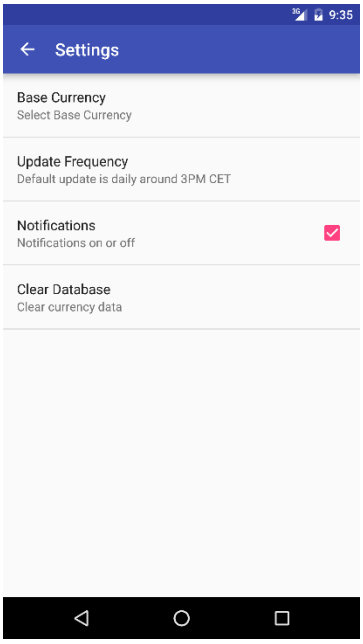
This is the chart activity. This will load a chart of the of the base currency vs the target currency rates. This will use the MPAndroidChart library https://github.com/PhilJay/MPAndroidChart.

Screen 3



This is the history activity. This activity is used to load historic rates of any currency since 1999. A spinner is used to select the currency to compare. An FAB button is used to load a date dialog, which then the user can select which date they would like the view the currency rates.

Screen 4



This is the settings activity. There are 3 list preferences and an extra preference. The first preference is to select the default base currency from a list. The second preference is to select the frequency the default base currency is updated (Daily or off). If daily is selected, one a day a service is run to check for an update of the base currency rate. The third preference is to be

able to turn on or off a pop notification of when the base currency has been updated. The forth item is a button which loads a dialog asking to clear out the database.

Add as many screens as you need to portray your app's UI flow.

Key Considerations

How will your app handle data persistence?

Data will be inserted into a SQLite Database. SharedPreferences class will also be used for a settings activity.

Describe any corner cases in the UX.

The UX will have 3 screens. Main activity for the base currency data, a history activity to retrieve historical exchange rates, a chart activity to chart out data, and a settings activity. A back button will be used for each activity to return to the main activity.

Describe any libraries you'll be using and share your reasoning for including them.

I will be using the MPAndroidChart library to create charts for the currency data.
https://github.com/PhilJay/MPAndroidChart

Next Steps: Required Tasks

This is the section where you can take the main features of your app (declared above) and decompose them into tangible technical tasks that you can complete incrementally until you have a finished app.

Task 1: Project Setup

Create a new project named Currency Exchange Rates
Include gradle dependencies for MPAndroidChart library.
Target latest sdk

Task 2: Implement UI for Each Activity and Fragment

List the subtasks. For example:

- Build UI for MainActivity/Fragment
- Build UI for ChartActivity/Fragment
- Build UI for History Activity
- Build UI for Settings Activity

Task 3: Create Build Variant

- Create debug folder
- Create release folder

Task 4: Create a Content Provider

- Create CurrencyProvider
- Create CurrencyContract
- Create CurrencyDBHelper

Task 5: Implement Utils & Constants classes

- Create methods for accessing network and retrieving JSON data (utils.class)
- Create all necessary public static final strings (constants.class)

Task 6: Implement MainActivity/Fragment

- Create Main Activity and Fragment classes
- Handle network error case
- Handle no data case (empty list)

Task 7: Implement Adapters

- Create Currency Adapter for listviews or recyclerviews
- Create Cursor Adapter to load data to listviews or recyclerviews

Task 8: Implement ChartActivity/Fragment

- Create Chart Activity and Fragment
- Use methods from library to implement chart

Task 9: Implement History Activity and DateDialogFragment

- Create History Activity
- Create Date Dialog Fragment
- Use FAB button to load DateDialogFragment
- Implement callbacks from dialog fragment to activity

Task 10: Implement Settings Activity/Fragment

- Create xml for preference fragment
- Create custom preferences

Task 11: Implement CurrencyService, Currency Receiver and JobScheduler

- Create CurrencyService as an intent service to check for updated data and download data
- Create a JobScheduler class to activate intent service on a daily basis
- Create CurrencyReceiver to insert downloaded data into database