Project 4 Task 2 – Currencyscoop, by Yufan Lu (AndrewID: yufanlu), Yuting Long (AndrewID: yutinglo)

Description:

Our mobile app will ask the user to enter two currency codes for conversion and the amount to convert.Then use Currencyscoop API to display the correct converted currency amount.

API name:
Currencyscoop

API url:
https://currencyscoop.com/api-documentation

Here is how my application meets the task requirements

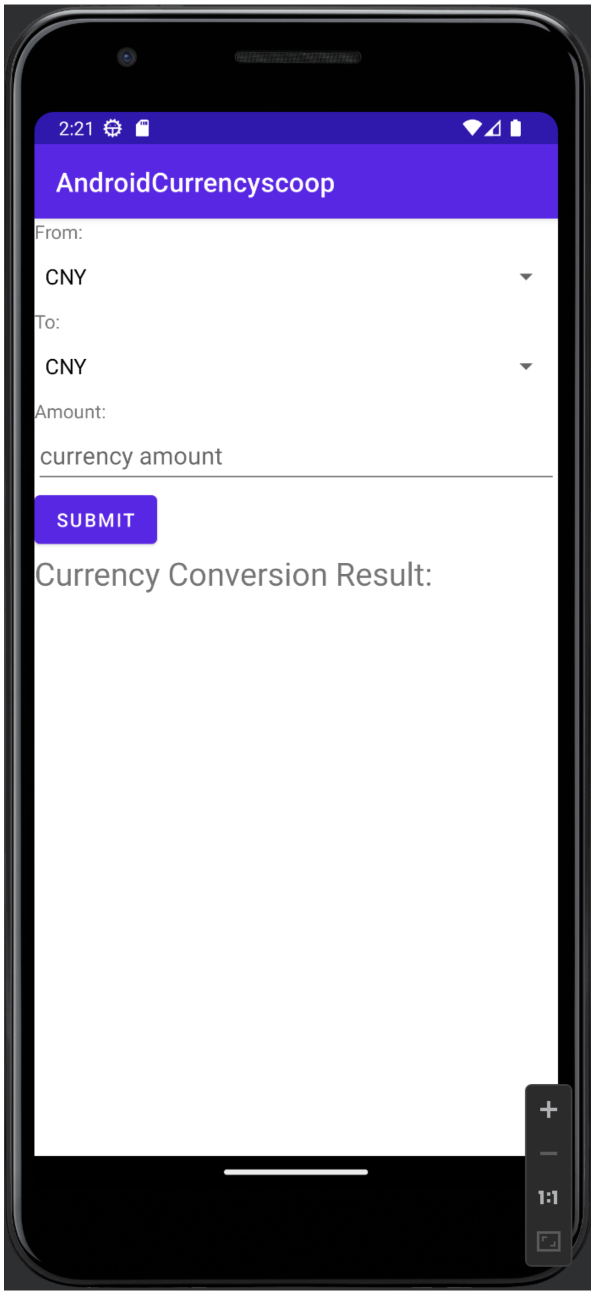
1. Implement a native Android application

The name of my native Android application project in Android Studio is: AndroidCurrencyscoop

a. Has at least three different kinds of views in your Layout (TextView, EditText, ImageView, etc.)

My application uses Spinner, TextView, EditText, Button. See content_main.xml for details of how they are incorporated into the LinearLayout.

Here is a screenshot of the layout before the response has been fetched.



From:

CNY

To:

CNY

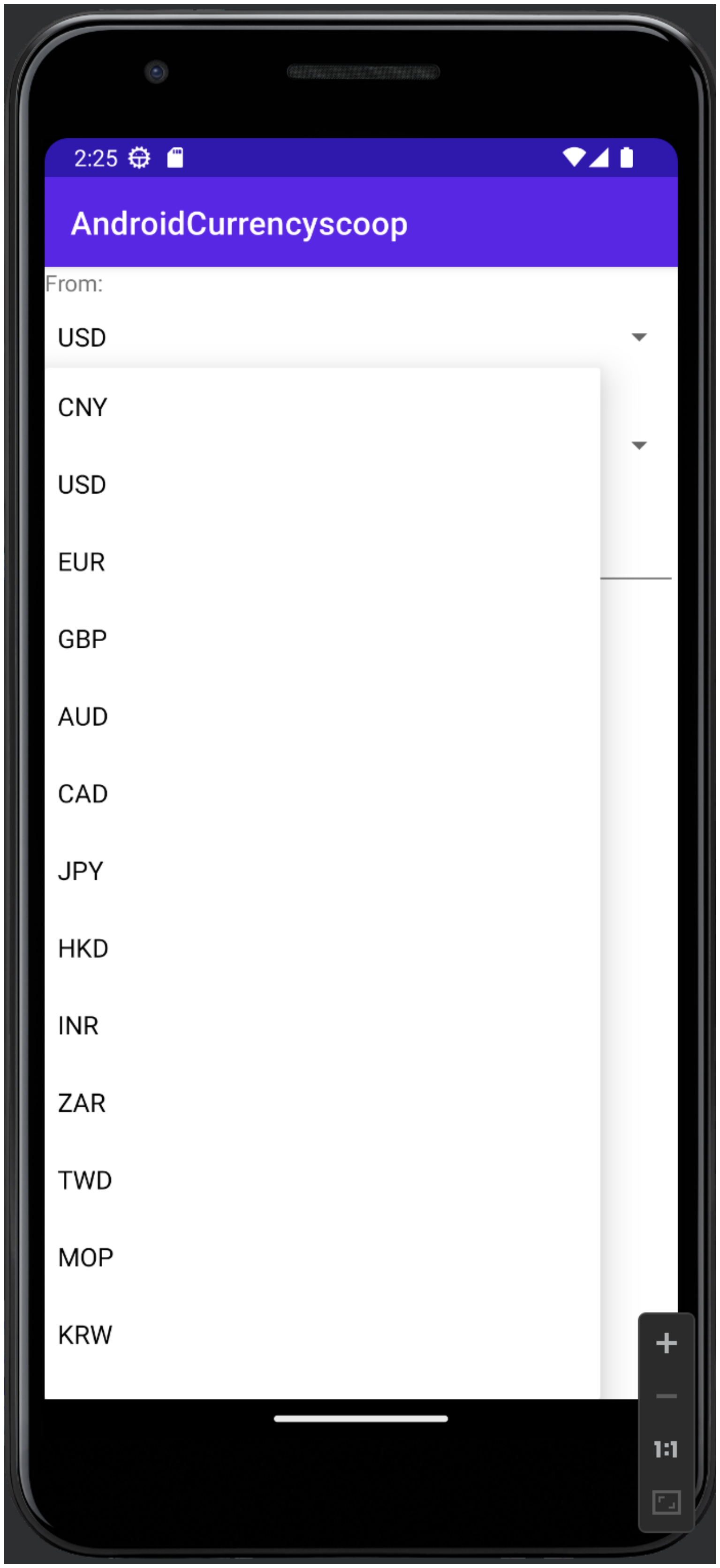
Amount:

currency amount

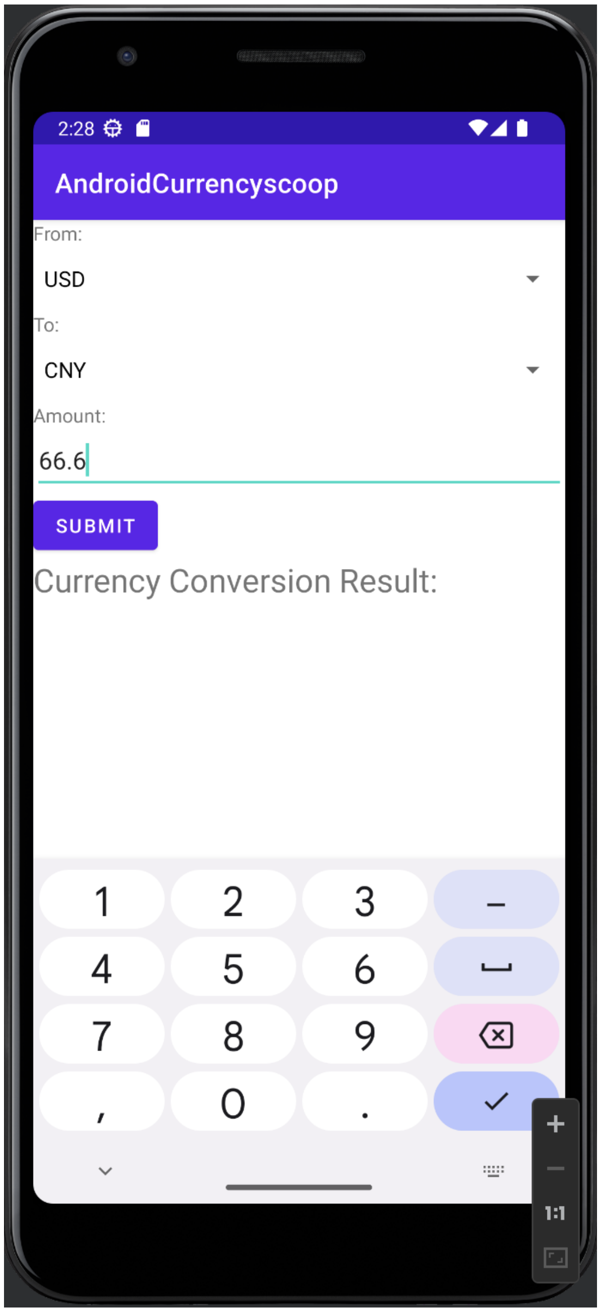
Currency Conversion Result:

b. Requires input from the user

Here is a screenshot of the user selecting a currency code (from/to) in the dropdown list.


2:25
AndroidCurrencyscoop
From:
USD
CNY
USD
EUR
GBP
AUD
CAD
JPY
HKD
INR
ZAR
TWD
MOP
KRW
1:1

Here is a screenshot of the user entering the currency amount he or she wants to convert.



c. Makes an HTTP request (using an appropriate HTTP method) to your web service

My application does an HTTP GET request in GetCurrencyConversion.java. The HTTP request is:

"https://powerful-wildwood-21341/getApiCurrencyscoop?currencyCodeFrom=" + currencyCodeFrom + "&currencyCodeTo=" + currencyCodeTo + "&currencyAmount=" +

currencyAmount where currencyCodeFrom and currencyCodeTo are the currency code selected by the user and currencyAmount is the amount entered by the user.

The search method makes this request of my web application, parses the returned XML to find the currency conversion result, and returns the result.

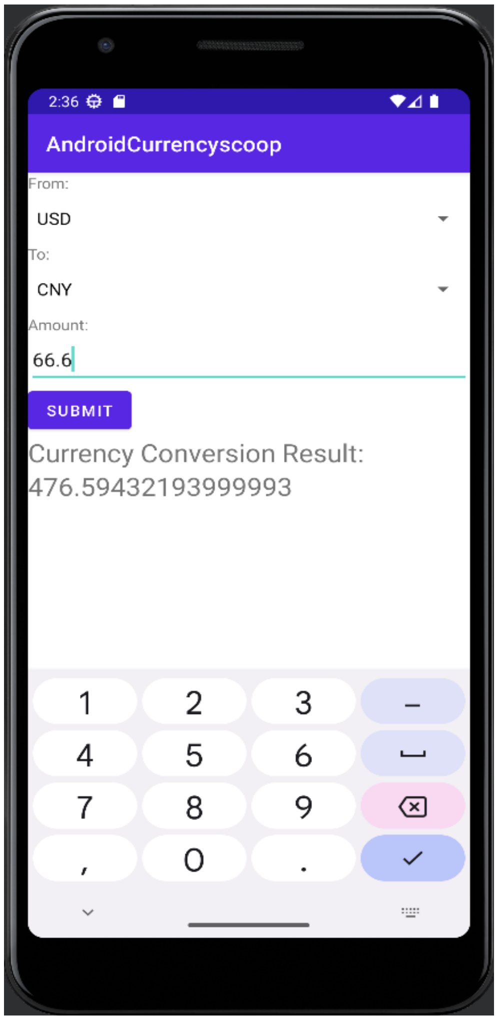
d. Receives and parses an XML or JSON formatted reply from the web service

An example of the JSON formatted reply is:

{"from":"USD","to":"CNY","amount":"66.6","result":"476.59432193999993"}

e. Displays new information to the user

Here is the screen shot after the response has been returned.


2:36

AndroidCurrencyscoop

From:

USD

To:

CNY

Amount:

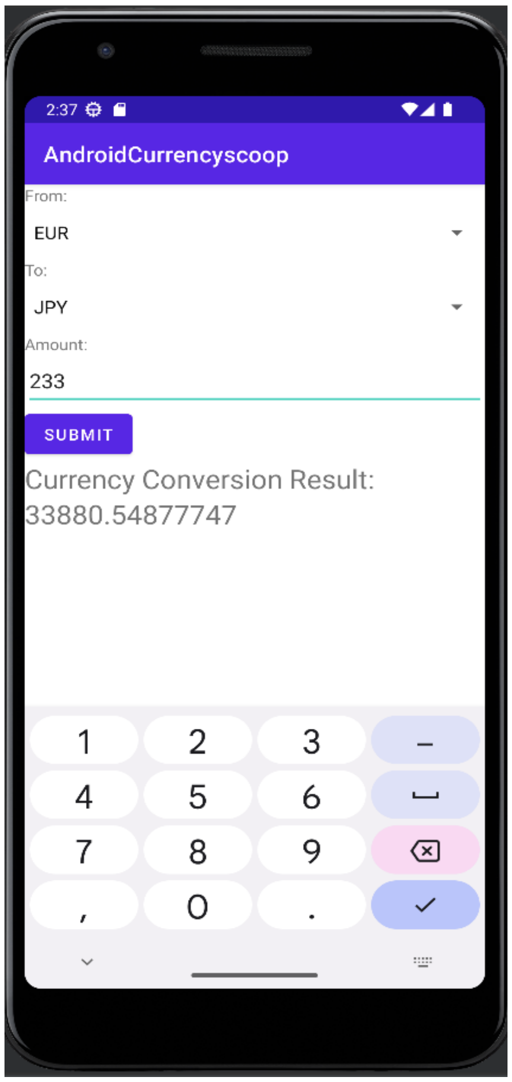
66.6

SUBMIT

Currency Conversion Result:
476.59432193999993

f. Is repeatable (I.e. the user can repeatedly reuse the application without restarting it.)

The user can type in another currencyCodeFrom, currencyCodeTo and currencyAmount and hit Submit. Here is an example of having typed in "EUR", “JPY” and “233”.



From:
EUR
To:
JPY
Amount:
233

Currency Conversion Result:
33880.54877747

2. Implement a web application, deployed to Heroku

The URL of my web service deployed to Heroku is: powerful-wildwood-21341

The project directory name is Project4/Currencyscoop.

a. Using HttpServlets to implement a simple (can be a single path) API In my web app project:

Model: CurrencyscoopModel.java

View: result.jsp

Controllers:

CurrencyscoopServlet.java

CurrencyscoopApiServlet.java

b. Receives an HTTP request from the native Android application

CurrencyscoopApiServlet.java receives the HTTP GET request with the argument "currencyFrom", "currencyTo" and “amount". It passes them on to the model.

c. Executes business logic appropriate to your application

CurrencyscoopModel.java makes an HTTP request to:

https://api.currencyscoop.com/v1/convert?from=" + from + "&to=" + to + "&amount=" + amount + "&api_key=" + API_KEY

It then parses the Json response and extracts the parts it needs to respond to the Android application.

d. Replies to the Android application with an XML or JSON formatted response.

We format the response to the mobile application in a simple Json format of my own design, here is an example:

{"from":"USD","to":"CNY","amount":"100","result":"715.60709"}

3. Handle error conditions - Does not need to be documented.

4. Log useful information - Itemize what information you log and why you chose it.

request_timestamp: the timestamp that the user sending the request, used to calculate latency

response_timestamp: the timestamp that the server replies, used to calculate latency

Currency_from: currency code that the conversion is from, used to analyze the top 5 countries.

Currency_to: currency code that the conversion is to, used to analyze the top 5 countries

Currency_amount: the amount the user wants to convert to cash, used to record the request

Status_code: the request status code, used to indicate the final status of the request

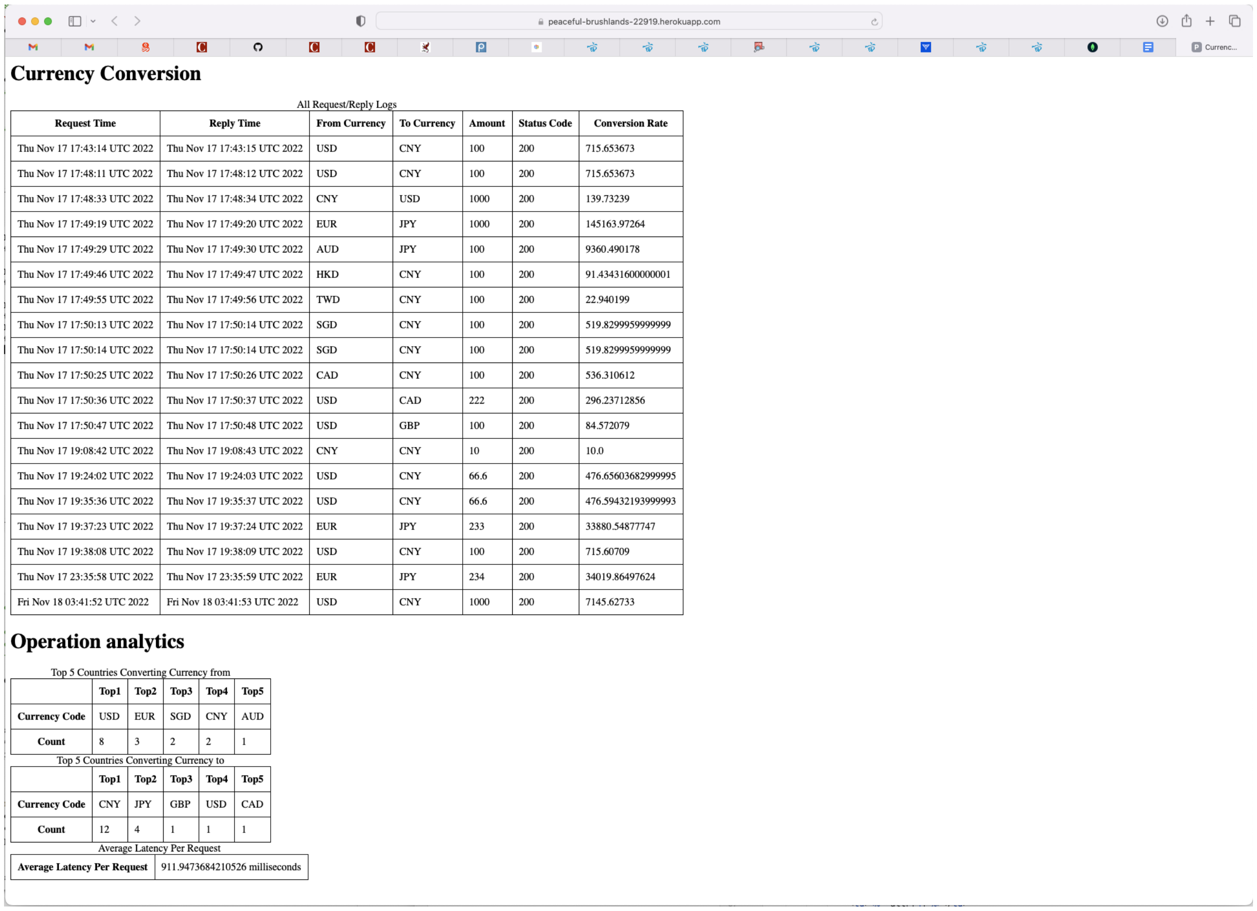
Reply_info: the result of the conversion rate, used to record the reply.

5. Store the log information in a database - Give your Atlas connection string with the three shards

```
mongodb://yufanlu1999:brobdingnagian@ac-qkzl6uw-shard-00-
00.xapckgu.mongodb.net:27017,ac-qkzl6uw-shard-00-
01.xapckgu.mongodb.net:27017,ac-qkzl6uw-shard-00-
02.xapckgu.mongodb.net:27017/test?w=majority&retryWrites=true&tls=true&authMe
chanism=SCRAM-SHA-1
```

6. Display operations analytics and full logs on a web-based dashboard - Provide a screen shot.

Here is the screen shot of operations analytics and full logs on a web-based dashboard.



Currency Conversion

All Request/Reply Logs

Request Time	Reply Time	From Currency	To Currency	Amount	Status Code	Conversion Rate
Thu Nov 17 17:43:14 UTC 2022	Thu Nov 17 17:43:15 UTC 2022	USD	CNY	100	200	715.653673
Thu Nov 17 17:48:11 UTC 2022	Thu Nov 17 17:48:12 UTC 2022	USD	CNY	100	200	715.653673
Thu Nov 17 17:48:33 UTC 2022	Thu Nov 17 17:48:34 UTC 2022	CNY	USD	1000	200	139.73239
Thu Nov 17 17:49:19 UTC 2022	Thu Nov 17 17:49:20 UTC 2022	EUR	JPY	1000	200	145163.97264
Thu Nov 17 17:49:29 UTC 2022	Thu Nov 17 17:49:30 UTC 2022	AUD	JPY	100	200	9360.490178
Thu Nov 17 17:49:46 UTC 2022	Thu Nov 17 17:49:47 UTC 2022	HKD	CNY	100	200	91.43431600000001
Thu Nov 17 17:49:55 UTC 2022	Thu Nov 17 17:49:56 UTC 2022	TWD	CNY	100	200	22.940199
Thu Nov 17 17:50:13 UTC 2022	Thu Nov 17 17:50:14 UTC 2022	SGD	CNY	100	200	519.8299959999999
Thu Nov 17 17:50:14 UTC 2022	Thu Nov 17 17:50:14 UTC 2022	SGD	CNY	100	200	519.8299959999999
Thu Nov 17 17:50:25 UTC 2022	Thu Nov 17 17:50:26 UTC 2022	CAD	CNY	100	200	536.310612
Thu Nov 17 17:50:36 UTC 2022	Thu Nov 17 17:50:37 UTC 2022	USD	CAD	222	200	296.23712856
Thu Nov 17 17:50:47 UTC 2022	Thu Nov 17 17:50:48 UTC 2022	USD	GBP	100	200	84.572079
Thu Nov 17 19:08:42 UTC 2022	Thu Nov 17 19:08:43 UTC 2022	CNY	CNY	10	200	10.0
Thu Nov 17 19:24:02 UTC 2022	Thu Nov 17 19:24:03 UTC 2022	USD	CNY	66.6	200	476.65603682999995
Thu Nov 17 19:35:36 UTC 2022	Thu Nov 17 19:35:37 UTC 2022	USD	CNY	66.6	200	476.59432193999993
Thu Nov 17 19:37:23 UTC 2022	Thu Nov 17 19:37:24 UTC 2022	EUR	JPY	233	200	33880.54877747
Thu Nov 17 19:38:08 UTC 2022	Thu Nov 17 19:38:09 UTC 2022	USD	CNY	100	200	715.60709
Thu Nov 17 23:35:58 UTC 2022	Thu Nov 17 23:35:59 UTC 2022	EUR	JPY	234	200	34019.86497624
Fri Nov 18 03:41:52 UTC 2022	Fri Nov 18 03:41:53 UTC 2022	USD	CNY	1000	200	7145.62733

Operation analytics

Top 5 Countries Converting Currency from

	Top1	Top2	Top3	Top4	Top5
Currency Code	USD	EUR	SGD	CNY	AUD
Count	8	3	2	2	1

Top 5 Countries Converting Currency to

	Top1	Top2	Top3	Top4	Top5
Currency Code	CNY	JPY	GBP	USD	CAD
Count	12	4	1	1	1

Average Latency Per Request

Average Latency Per Request	911.9473684210526 milliseconds