Project 4 Task 1 - Currency Conversion
By Haoran Chen (Andrew ID: haoranc3)

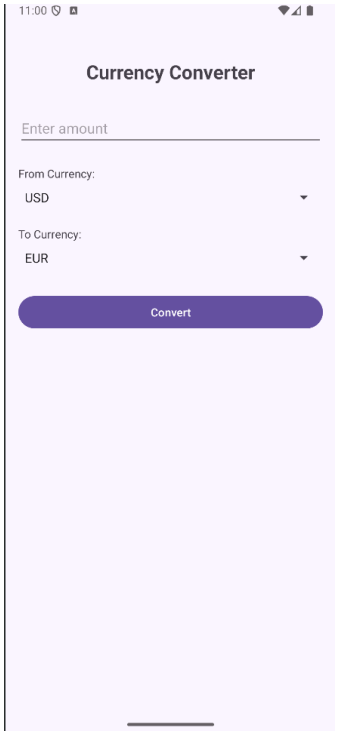
Description:
My application is a currency converter build using Forex API

Here is how my application meets the task requirements

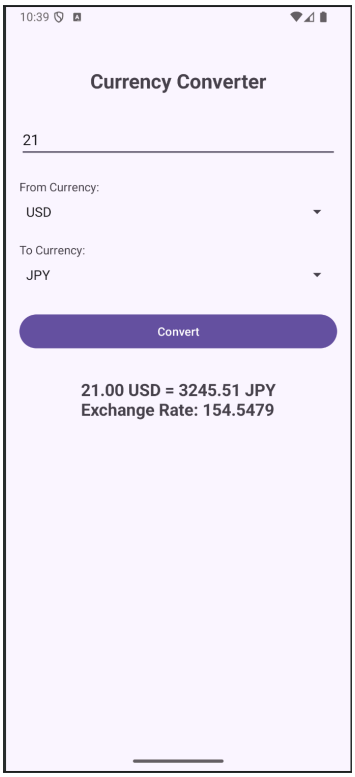
1. **Implement a native Android application**
    a. Has at least 3 different kinds of views in my layout
    (TextView, EditText View, DropDown Selection)

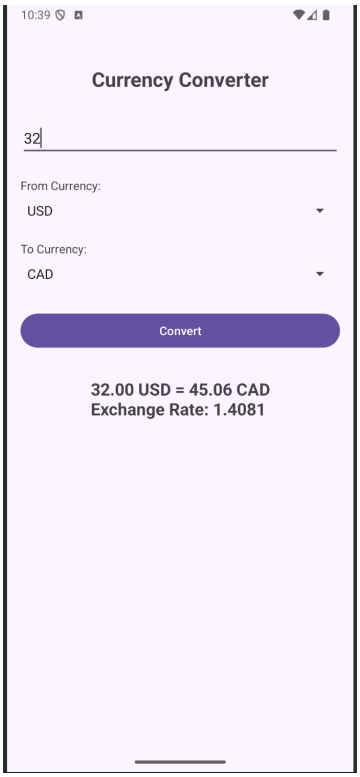
My application uses TextView, EditText, Button, and Spinner.
See res - layout for details

Here is a screenshot of the layout before we enter an desire number and choose currencies



Here is a screenshot of the user input for a currency and chooses currencies for exchange





Makes a HTTP request (a get method) to the server side

My application does an HTTP GET request in api - ForexApiService
The HTTP requests is:

{base_url}/convert?from=USD&to=EUR&amount=100

This method send a GET request to server side which communicates with Forex API, fetches the LIVE result, and return the rate of conversion between two currencies.

An example of the reply is:

{ "from":"USD", "to":"EUR", "amount":22.0, "result":20.875799999999998, "rate":0.9489 }

2.Implement a web application, deployed to GitHub Codespace The URL of my web service is:

NOTE: for some reason, I could not figure out why, my GitHub might shut down the running server, if you want to test it using the URL + request body, **you might need to + new code space** under the corresponding repository on my GitHub.

```
https://expert-fiesta-7p574qgrj7wcx6j7-8080.app.github.dev/
```

a. Using HttpServlet to implement a simple API In my web app project: Model: LoggingModel.java - Contains data structures for forex conversion and logging View: dashboard.jsp - Displays analytics and monitoring data Controller: ForexServlet.java - Handles currency conversion requests

b. Receives an HTTP request from the mobile application ForexServlet.java receives the HTTP GET request with three parameters:

- "from": Source currency code (e.g., USD)
- "to": Target currency code (e.g., EUR)
- "amount": Amount to convert

c. Executes business logic appropriate to your application The ForexServlet makes an HTTP request to:

https://v6.exchangerate-api.com/v6/[API_KEY]/pair/USD/EUR

It then parses the JSON response and calculates the converted amount based on the current exchange rate.

d. Replies to the application with a JSON formatted response: json


```
{
   "from": "USD",
   "to": "EUR",
   "amount": 100.00,
   "result": 92.15,
   "rate": 0.9215
}
```

3. Handle error conditions:

- Missing parameters
- Invalid currency codes
- Invalid amount format
- External API errors
- MongoDB connection issues All errors return appropriate HTTP status codes and JSON error messages.

4. Log useful information: We log the following information for analysis:

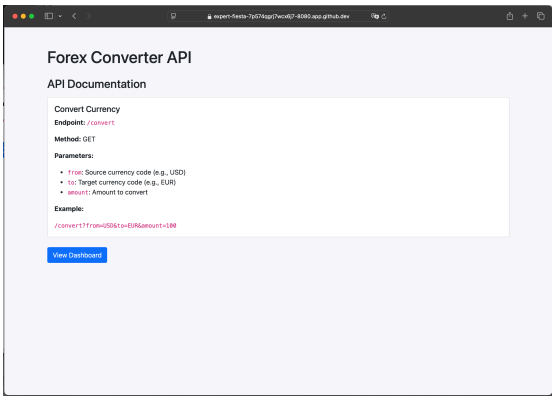
- Request timestamp
- Source/target currencies
- Amount to convert
- Exchange rate used
- Response time
- Success/failure status
- Error messages (if any)
- Client device information This information helps track usage patterns, monitor performance, and debug issues.

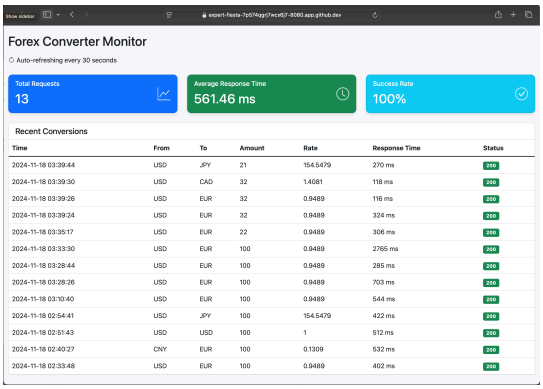
5. Store log information in MongoDB: MongoDB Atlas connection string:

   mongodb+srv://haoranc3:Djk7iJRjIFTmol0l@cluster0.ctex7.mongodb.net/?retryWrites=true&w=majority&appName=Cluster0

6. Display operations analytics and full logs on web-based dashboard: The dashboard (dashboard.jsp) shows:

- Total number of requests
- Average response time
- Success rate
- Real-time conversion logs
- Currency pair usage statistics





This implementation:

- Uses Jakarta EE Servlets
- Integrates with ExchangeRate API
- Stores logs in MongoDB Atlas
- Provides real-time monitoring(update in 30 seconds)
- Supports error handling and debugging
- Includes comprehensive request logging