Currency exchange

Your task is to create a currency exchange widget. The widget is composed out of a web user interface (REACT) backed by an API (NodeJS).
The API accepts queries from API clients asking to get a quote for exchanging money from one currency to another.

API request interface

Method: GET; Request URL: <BASE_URL>/api/quote

from_currency_code	String, 3 letters currency code. Currency to convert from. Examples: USD, EUR, ILS
amount	Integer. The amount to convert in cents Example: 100 (1 USD)
to_currency_code	String, 3 letters currency code. Currency to convert to. Examples: USD, EUR, ILS

Supported currencies are USD, EUR and ILS.
The API calculates the total amount expected in the to_currency_code according to an exchange rate provided by a 3rd party.

API response interface

Response body: JSON with the following properties.

exchange_rate	Decimal, the offered exchange rate. Up to 3 decimal digits Examples: 1.234, 4.5, 2
currency_code	String, 3 letters currency code. The currency of the amount. Examples: USD, EUR, ILS

amount	Integer, the expected amount in cents. Calculated as request amount * exchange_rate Comment: You can select a rounding policy

3rd party exchange provider

Exchange rates are fetched from https://api.exchangeratesapi.io.
Issuing a GET request to the following URL will return the latest exchange rates for several currencies when the base currency is the United States Dollar (USD).
You can try it in a browser window to see the response format

https://api.exchangeratesapi.io/latest?base=USD

For performance and cost reasons, exchange rates are to be cached for a duration of 10 seconds during which any request for a specific “from_currency_code” will be served from the cache (will not go out to https://api.exchangeratesapi.io).
For the purposes of this exercise, no need to use a database to store cache.

. The algorithm:

1. Get conversion details from request
2. Check for exchange rate in cache. If not found, fetch latest exchange rate from 3rd party
3. Calculate amount according to exchange rate

Example request and response

curl
http://localhost:3000/api/quote?from_currency_code=EUR&amount=100&to_currency_code=I

LS => { “exchange_rate”: 4.5, “currency_code”: “ILS”, “amount”: 450 }

Exchange widget

Now it’s time to create a user interface, a small single page application for displaying our

exchange rates.

Layout

Rate section
- Base currency dropdown
- Quote currency dropdown
- Base amount input

Results section:
- The received rate
- The expected amount

Functionality

- At first, only the rate section is visible.
- User can select from Base and Quote currency dropdowns and enter an amount in the Base amount field. Once all fields are populated **and valid**, a request to get the latest quote is made.
- Once results are returned, the results section is shown. The conversion rate and the expected amount are presented to the user.
- Changing quote:
    - After all fields are populated, the user can change any input in the rate section (amount or currencies). Such change triggers an automatic new quote request and updates the results section with the new rate and amount

Important notes

- UI experience needs to be compatible for both Desktop and mobile
- Validation errors are to be presented when applicable
- Consider UX, network calls may take time. User should be aware that something is happening