Example API Call-

A request/response to/from the api.frankfurter.app API might look like the

GET https://api.frankfurter.app/2020-07-07?

{"amount":1.0."base":"NZD"."date":"2020-07-07", "rates": ("AUD": 0.94212, "USD": 0.65411)}

When converting your line amounts, please pay attention to the direction of the exchange rate you've retrieved from api.frankfurter.app e.g. USD -> NZD vs NZD -> USD. Each line amount should be converted from the line currency to the invoice currency.

Valid Currencies are ALID Australian Dollar

 On input error, the relevant input field should be highlighted BGN Bulgarian Lev as red. (see example below) BRL Brazilian Rea CAD Canadian Dolla Other errors should be displayed as alert. CHF Swiss Franc CNY Chinese Renminhi Yuan DKK Danish Krone -700 EUR Euro GBP British Pound HKD Hong Kong Dollar HUF Hungarian Forint IDR Indonesian Rupial ILS Israeli New Shegel (1) Error: fetching exchange rate. INR Indian Rupee ISK loelandic Króna .IPY .lananese Yen KRW South Korean Won MXN Mexican Peso Special validation rule MYR Malaysian Ringgi Please ensure each line item with in ONE invoice is NOK Norwegian Krone NZD New Zealand Dollar PHP Philippine Peso unique PLN Polish Zloty RON Romanian Leu SEK Swedish Krona Intel Core i9, 700, USD SGD Singapore Dollar THB Thai Baht Intel Core i7, 700, USI TRY Turkish Lira USD United States Dollar ZAR South African Rand Intel Core i9, 800, USD Intel Core i9, 900, USD Intel Core i9, 700, USD



×

Your objective is to create a multi-currency involcing application

The application should be implemented as a Next. is application.

The application will be required to calculate the correct invoice totals. We expect the correct result is always shown on screen by

the application. The step to calculate total for each invoice is

2. Calculate each line amount in the base currency and

3. Display the invoice total in the required area

Relevant totals are summarised in Totals section

1. Use the frankfurter.app API to retrieve the exchange rate for

which uses frankfurter.app as the exchange rate input.

The expected interface is shown on the right.

using Material UI as the UI library.

each of the currency pairs.

accumulate the total

On success

On error:

- 1. We will check that you have correctly followed the instructions. After reviewing your code, test your application extensively against a set of predefined scenarios to ensure your application has catered for additional edge cases.
- We will score your submission against:
 - Code Design Quality

Evaluation Criteria

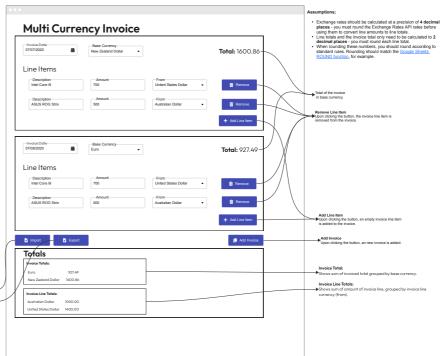
- Error Handling
- Code Performance Security
- Operability

Documentation

We expect a solution can be completed within 1-3 hours total development time

Please do not attempt to add additional features we have not asked for. We want a working solution which addresses our evaluation criteria nothing more.

If you do not believe you can complete a solution in under 3 hours, please let us know. We do not "police" this time limit - it simply exists to be respectful of your time.



Paste into / Copy from the JSON representation of the invoices here. See example input for detail.