



## Full-Stack developer homework assignment

Dear candidate,

Your task is to build a basic FX (Foreign Exchange) positions data webpage.

The screen will include a table that displays the updated FX positions based on the latest currencies rates and a button to download the detailed table of the positions.

### General Definitions:

- **Financial Unit** - a basic unit that instructs the applicable trading options for a specific trader.
- **Position** - amount of a security, commodity or currency that is owned (a long position) or borrowed and then sold (a short position) by an individual, institution or dealer (you don't have to understand exactly what a position is - just associate each position to the correct financial-unit).

### Task details:

1. Implement the webpage using reactJS.
2. Implement the server side using a language of your choice.
3. You should fetch the latest currency rates (list of currency rates vs USD) from an external free API.
4. Two JSON files are attached to this task:
  - a. positions.json
  - b. financial\_unit.jsonyou can read their content from the file or copy their content into your server code base.
5. The goal of this task is to show the financial\_units and their associated positions in a react table, while you calculate the notionalValues for each position by using each one of the position's currencies.

**Rate** column - consists of the currency rates vs USD received from the external API.

**Currency** column - consists of the currency code (3 letters such as JPY, EUR, CHF).

**Calculated Value** column - consists of the notionalValues of the currency in USD terms.

An example for the required table:

Financial Name	Unit	Notional Value	Rate	Currency	Calculated Value (in USD)
Fin Unit 1		5000	1.1748	EUR	5874
Fin Unit 1		2000	0.0088	JPY	17.5925
Fin Unit 2		300	1.0147	CHF	304.4049

6. After the table from #5, please also generate a table that shows aggregated “Calculated Value (in USD)” per Financial Unit Name.
7. The data should be refreshed automatically every 10 seconds.
8. You should create a button on this screen - “Export to Excel”. This button will allow a user to download the table from #5 on the screen into a CSV file.  
\*\*\* It is possible to replace the export to excel mechanism with invocation of writing the tables details to a file (on the server side).
9. Loading spinner – please add a loading spinner of your choice. The spinner will be displayed on top of the rendered screen for 3 seconds and then disappears.
10. Bonus: create currency exchange calculator at the top of the screen to assist the users for further trades.

**Currency:** drop down list of all currencies you received from the external API (CHF, AUD, CAD, JPY, EUR, ...).

**Notional Value:** field the user will fill with the amount of the currency

**Calculated Value:** the value of the currency in USD

EUR	1000	1187.000
Currency	Notional Value	Calculated Value (in USD)

**Key points:**

1. **Don't use any sockets technology.** Use regular HTTP/Web requests for communication between client and server.
2. Design - we expect a basic design. Needless to say, you'll get bonus points if the design looks neat.