# AdAlpha technical task write up

## Assumptions/Requirement gathering

After reading the task requirement document I had a few questions, which I felt I needed to go back to the customer to clarify prior to planning any potential solution, these questions were:

1. I understand that there are this list of assets and the total current value is £220,000 and the percentages for each asset of that value. Where I am struggling is the order sheet instructions, for the BUY instruction it states it wants to know how many units you want to purchase of a particular asset, I would interpret that the buying of units of assets would increase the £220,000 total value but how do I know how much a unit of an asset is valued at?

1. Also I am not too sure what the difference between BUY and INVEST is? same as I am not sure what the difference between SELL and RAISE are? Other than one is in units and one is in currency.

After gathering an understanding of what each instruction meant and information on where asset prices were listed I felt I understood the task.

## Description of solution

When planning the solution, I tried to include as many technologies/skills off the job advert to show a willingness to learn and try new technologies even if it’s not something I am familiar with.

The UI provides two tables, one of the investor’s portfolio and one of the investor history. I implemented all for instructions (Buy, Invest, Sell, Raise) and there are action buttons for each of these in the portfolio column.

When these are triggered in the UI they call a POST request to the API which will then get the asset price by scraping the financial times website, this price is just included in the investor history for the BUY & SELL instruction but for the RAISE & INVEST instructions it is used in the actual calculation of the units. Once the calculation has taken place the units will be updated in the portfolio table for the investor and a history item will be added to the instructions table.

When an INVEST OR RAISE instruction is used the price is scraped as mentioned above and depending on what currency code the investor has selected as the invest or raise amount, will then call a third party api to get the current exchange rate from that currency to GBP as the total units to add or remove from the portfolio are calculated using GBP.

e.g RAISE instruction of $50 from isin x will, (50 \* (exchange rate to GBP) ) \* asset price will give the total amount of units to remove from the investors profile.

I created a custom logger which uses the go Elasticsearch client to post info/error logs to Elasticsearch, I feel this gives the users an easier way to view logs rather than going into the actual docker container to view them.

I also used a web scraping library to get the current asset price from the Financial times website, I would have preferred to use a third party api but I couldn’t find a free once which provided the asset prices required. There were issues with constantly scraping the financial times website so I implemented a map which updated the asset prices once every hour.

## Self-evaluation

Considering nearly all technologies used in the solution are new to me I am quite pleased with what I produced.

As it was the first time I have used Go in a project other than doing a course which is a lot different to actually writing a service I spent a lot of time on the backend and getting that right. This actually allowed me to further increase my knowledge of the language, and considering it is something I have not done previously I was quite pleased with what I produced. I had to learn how to test in Go which was a learning curve as it is very different to testing in Java.

I have also not used VueJs before so again another learning experience. UI dev is not something I have done in a while but is something I enjoy when I do it so is definitely something I would love to get better at. There are a lack of tests on the UI as I spent most of the time on UI learning VueJS. Once I got my head around how it compiled I started to enjoy writing it.

Elasticsearch, Kibana postgres, sql and docker are all technologies I have used before so had no issue in using them.

If I were to do the task again, the majority of what I would change would be on the UI front. I would want it to first of all be a lot nicer visually and I would want to include unit & e2e testing. Unit and e2e tests on a JS UI are something I have never had experience in, along with Typescript so I stuck to writing vuejs in JS as there were so many new technologies to me on this task I would have struggled to learn VueJS and Typescript simultaneously.

The main thing I got out of doing the task is how much I enjoyed writing Go and has confirmed is something I would love to do in my work life, what I have produced I’m sure there are better ways I could have done things but that will come with more experience with the language.

I hope in providing a working solution in a lot of new technologies I have showcased my willingness and passion to learn new things and constantly improve as a developer.