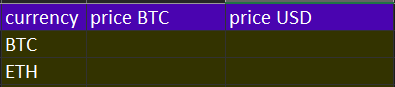
|  |  |
| --- | --- |
|  | **Szymon Babczyński**  Contact: szybabczynski@gmail.com  Github: https://github.com/rvnlord/ |

**Cryptocurrency Portfolio in Excel**

I bet everyone at some point heard about websites or applications offering fancy ways of displaying, summarizing and organizing your crypto portfolio. Example of such projects are *Cryptocompare* or *Blockfolio*. Inventions such as these are often welcomed with mixed feelings and conspiracy theories about the true purpose of these applications. Due to that fact in today’s article we are going to learn how to build our very own portfolio in Excel and utilize a live data feed.

First thing we need to do is create 2 sheets in our new blank Excel document, add the appropriate column headers to the first sheet and mini utility table to the second one:

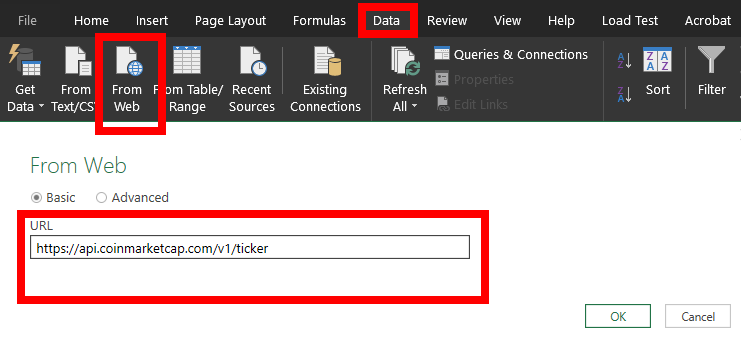



Next thing we require is connection to the data source that can provide us with live price changes. We are going to use *Coinmarketcap* public *Web API* for this purpose as it is straightforward and one of the most popular.

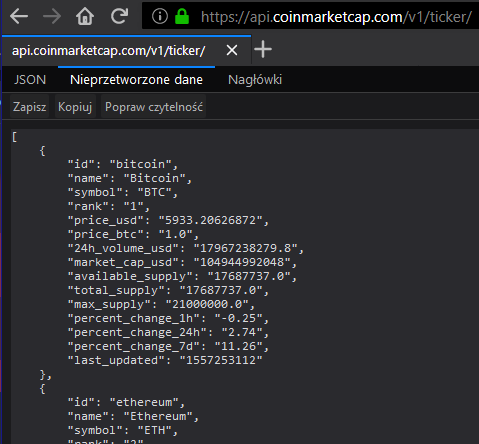
In order to add a new connection: select *data* tab, *from web* and paste the following connection string into the popup window:

https://api.coinmarketcap.com/v1/ticker

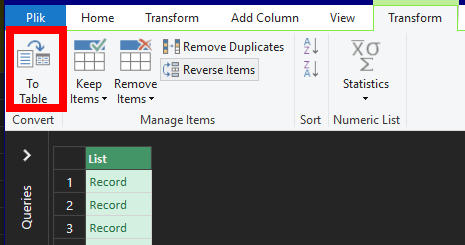
The process should look like this:



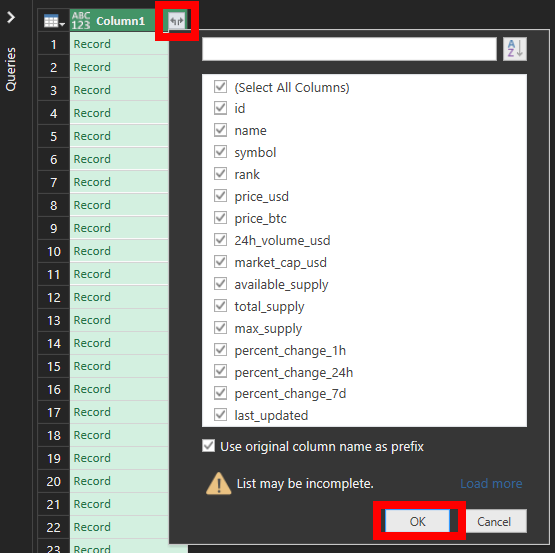
As soon as we confirm creation of our new data source, a configuration sheet should appear on the screen, but to know what to do, we need to see how the data produced by our *GET* request looks like first. To make it simple, let’s just paste the connection string into the browser and take a look at the results.



Result is basically an array of *JSON* objects representing top 100 cryptocurrencies by volume. In Excel however we work with tables by design, so in order to convert our array into a table we have to select *Transform* tab and *To Table* option.

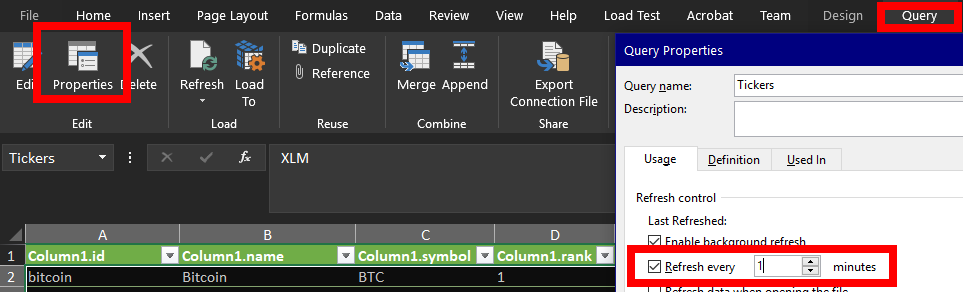


Then we have to expand the columns so we can later query the properties we are interested in.



The last thing to do is choose *Close & load* option in the upper left corner of the workspace to create a new worksheet with our transformed data. Let’s rename the worksheet to *Tickers* for easier referencing.

At this point we have the data but it isn’t updating automatically, in order to enable this feature we need to select *Query > Properties* and enable *Refresh every x minutes* option.

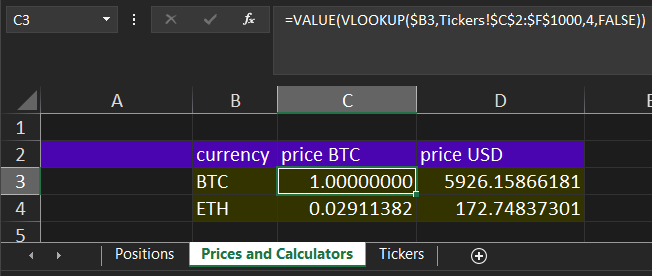


To fill our utility table next, we are going to make a formula that takes the price in appropriate quote cryptocurrency by symbol. So basically we have to paste the following strings in *BTC* row and drag the right corner of both cells to the Ethereum row in order to copy the logic.

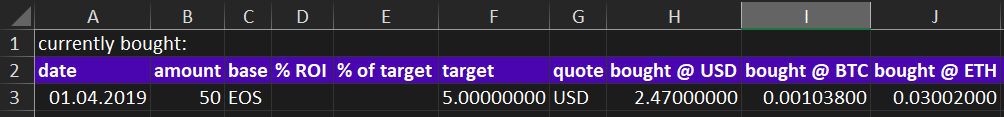
=VALUE(VLOOKUP($B3,Tickers!$C$2:$F$1000,4,FALSE))

=VALUE(VLOOKUP($B3,Tickers!$C$2:$F$1000,3,FALSE))

The following picture shows the outcome of our actions. The table should also update every x minutes.



For the purpose of this tutorial let’s assume that we bought 50 EOS at 2.47 USD at the beginning of April 2019.



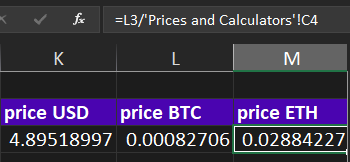
Let’s write some formulas that will calculate changes to our positions on the fly. First, we need to find price in at least one quote cryptocurrency, the others might be calculated from it. Alternatively we could take all the values from the data table, but in this example we will pursue the first option. Computation of price in BTC will look like the one we have seen before in our utility table:

=VALUE(VLOOKUP($C9,Tickers!$C$2:$F$1000,4,FALSE))

Price for both USD and ETH is then just a simple arithmetic operation by using our utility table:

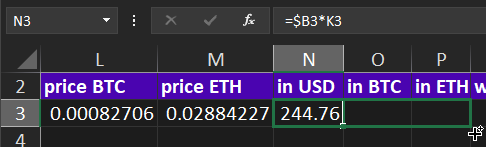
=L3\*'Prices and Calculators'!D$3$

=L3/'Prices and Calculators'!C$4$



Calculating amounts we have and expressing them in quote cryptocurrencies consists multiplying each amount by the appropriate price. So we simply have to paste the multiplying formula into the first *amount* cell and drag it to the others:

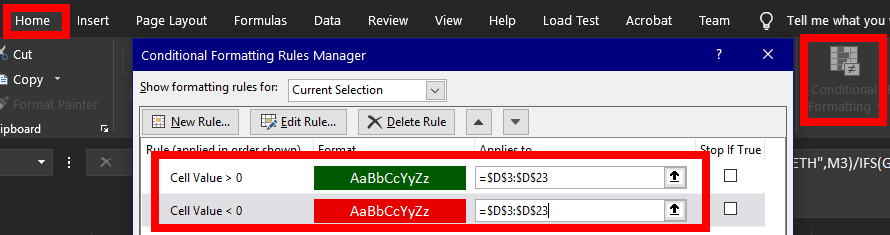
=$B3\*K3



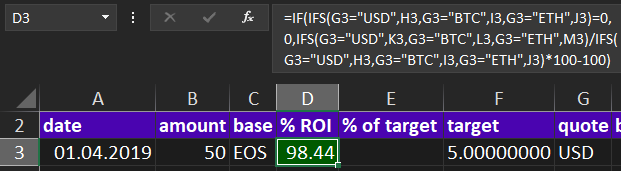
To also have the return of interest displayed in our table we need to take the current price in chosen quote cryptocurrency, divide it by original price (also in chosen cryptocurrency) and take the percentage.

=IF(IFS(G3="USD",H3,G3="BTC",I3,G3="ETH",J3)=0,0,IFS(G3="USD",K3,G3="BTC",L3,G3="ETH",M3)/IFS(G3="USD",H3,G3="BTC",I3,G3="ETH",J3)\*100-100)

Let’s conditionally format the *ROI* column as well, so it displays in green when we are earning money and in red when we are losing. To do this we need to select *Conditional formatting* and create two rules of type *format only cells that contain*.

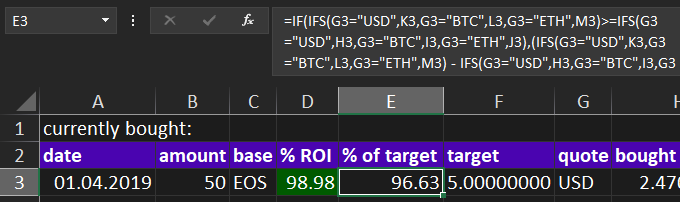


The result should look like this:



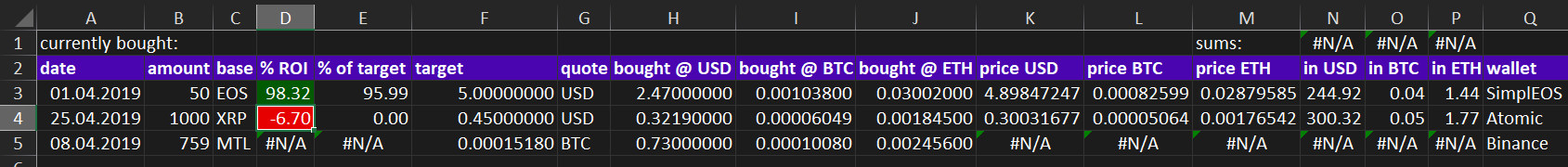
It would also be nice to know when exactly we should exit our position, *% of target* column serves exactly that purpose. In the following formula we basically compute percentage of our target in case if we are in green.

=IF(IFS(G3="USD",K3,G3="BTC",L3,G3="ETH",M3)>=IFS(G3="USD",H3,G3="BTC",I3,G3="ETH",J3),(IFS(G3="USD",K3,G3="BTC",L3,G3="ETH",M3) - IFS(G3="USD",H3,G3="BTC",I3,G3="ETH",J3)) / (F3 - IFS(G3="USD",H3,G3="BTC",I3,G3="ETH",J3)) \* 100,0)



It is a good idea to also conditionally format the above column, but at this point you know the drill. The difference here would be that condition should show the cell as green when the value is >=100.

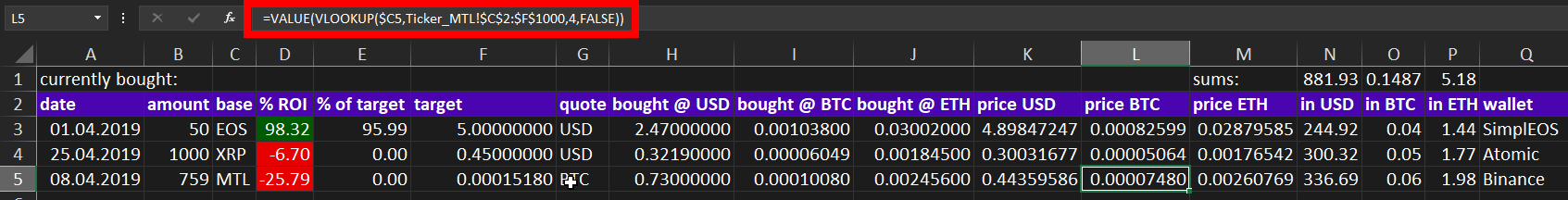
Well, it looks like our work should be completed now, but when we start adding assets into the table we immediately notice that something is wrong:



The cryptocurrencies that are not in top 100 on *Coinmarketcap* won’t be found in our data source, so we need to query them separately by adding a new data source and specifying cryptocurrency *Id*.

https://api.coinmarketcap.com/v1/ticker/metal/

The process is exactly similar to the one we used while adding top 100 tickers. We are converting the results into a table and expanding the columns, then we are loading the table into a new spreadsheet. After that, all that remains is to modify the formula that calculates the BTC price to include the new spreadsheet data instead of the old one.



That’s it, now everything should work properly and display correct values. Congratulations, you have just built your very own portfolio summary and you can modify it as you see fit.

