

Eikon Python API Wrapper

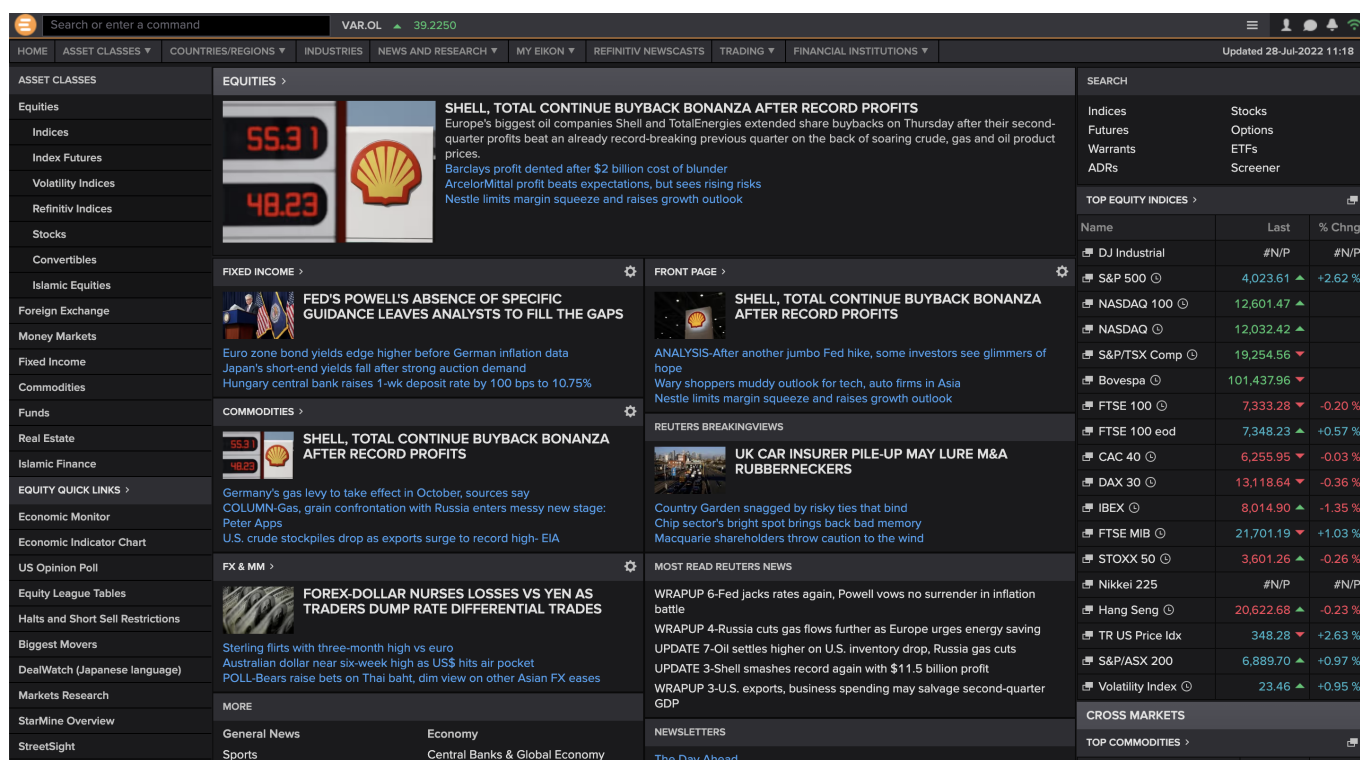
Project overview

Welcome to the future of writing a quantitative Project and Master's thesis at Institute for Industrial Economy (or whatever else you would like to use these tools for, feel free). This repository is created by two former students at Indøk, who, during our work on the master's thesis, created the precursor to the code to make the process of collecting large amounts of financial data as easy as possible. In this repo, we have tried to provide a generalized system that allows for very easy access to the most-used financial data (stocks, bonds, commodities), examples, and documentation. In short, this is the project we wished existed when we embarked on our Master's journey a dark and cold March night in 2022.

This project assumes some basic knowledge of Eikon. For a in-depth guide to using Eikon together with the Excel plugin to extract bond data, see [Fosse \(2021\)](#). This guide goes in deep detail on using the Excel plugin, and could be wise to peruse when working with Eikon.

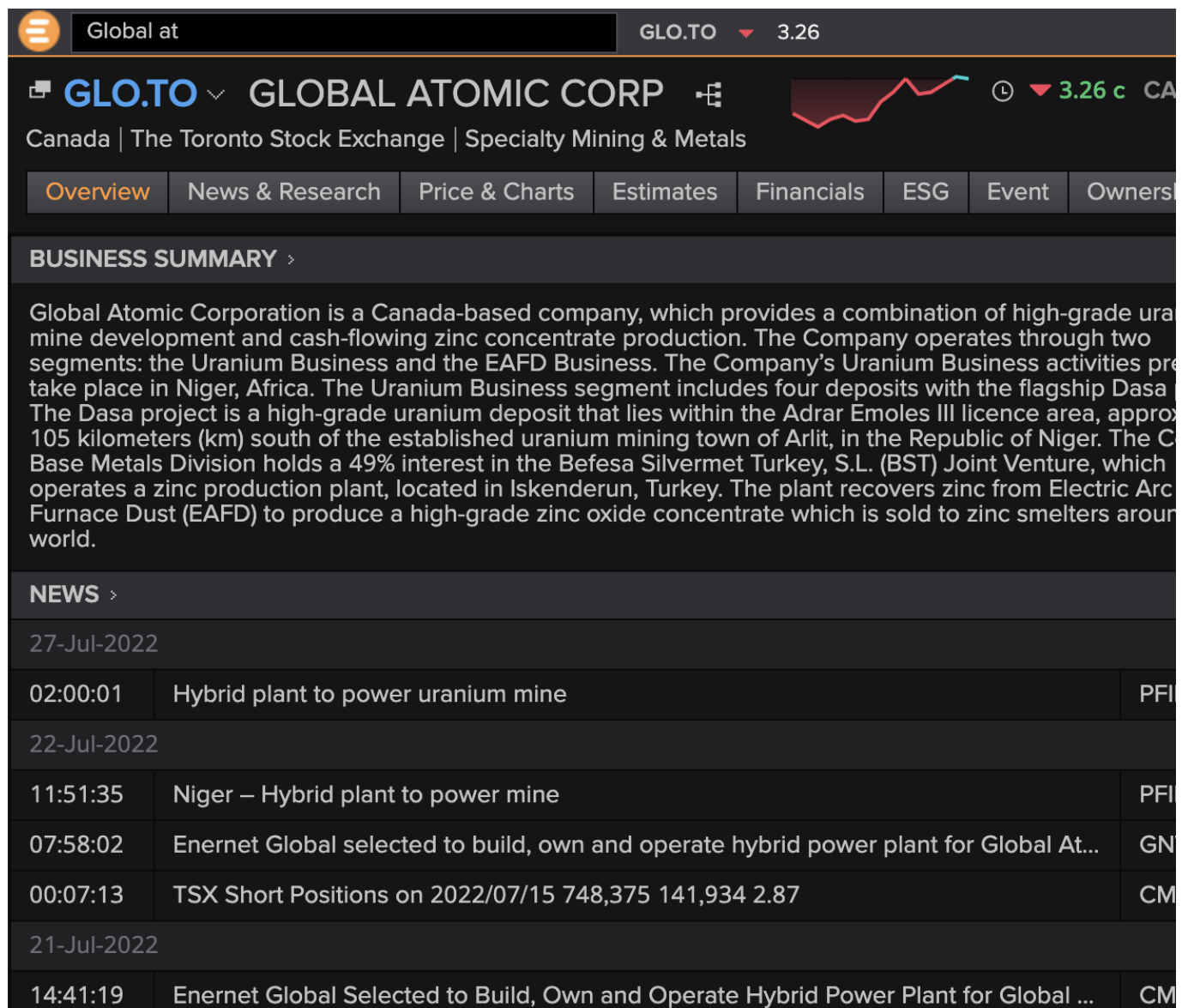
Introduction to Eikon

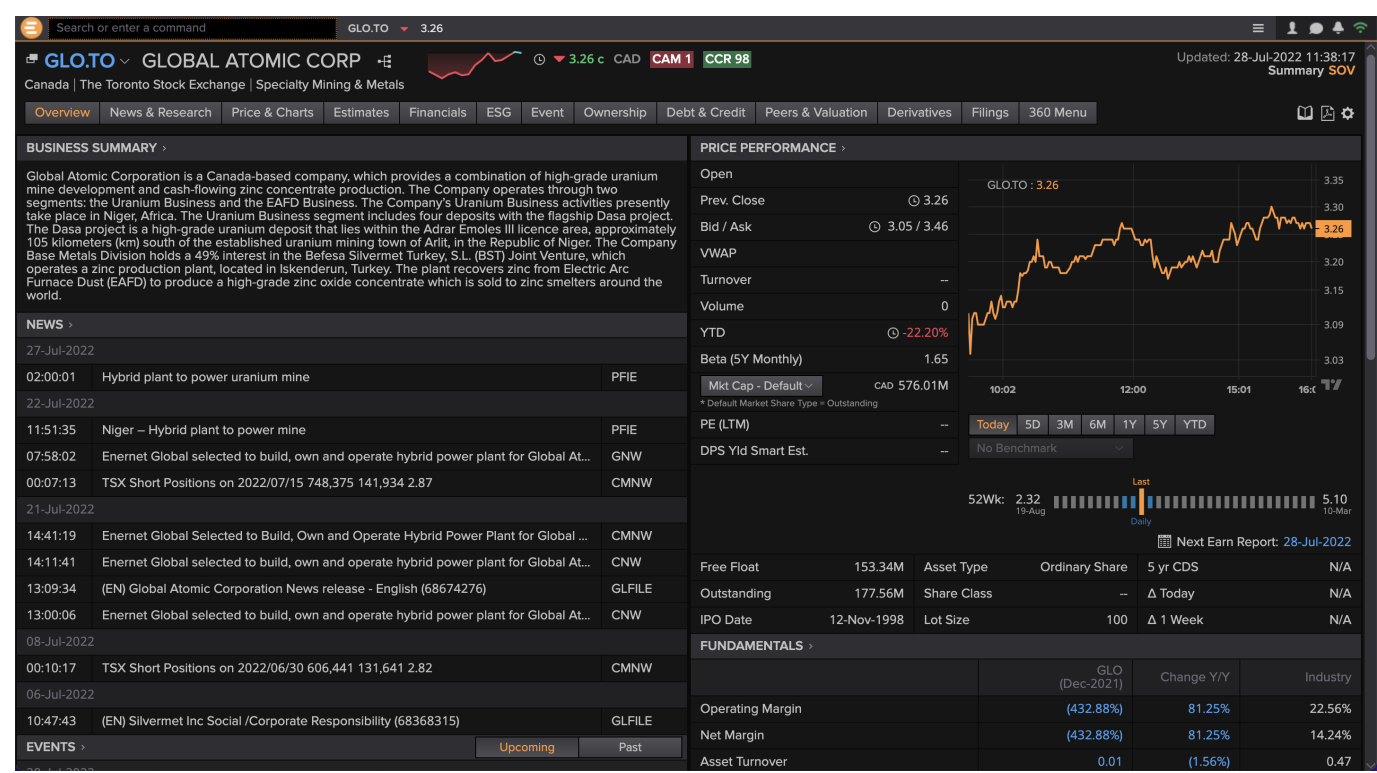
Refinitiv Eikon (Eikon) is a solution made for financial professionals to provide wide array of financial information for several asset classes and geographies. Detailed and up-to-date information is available on equities (stocks), bonds, commodities, forex, and other macro financial variables. For comparison, Eikon is similar to the well-renowned Bloomberg Terminal, and serves the same purpose, only to a slightly lesser extent. However, the present project seeks to bridge the gap somewhat.



You can access the data through several avenues, e.g., an iPhone application, a Windows desktop application, a web interface, and an Excel plugin. The main hub of research usually centers around the Widows application, often termed the "Terminal." Here, one can explore the data the system provides in a visual interface. For example, one can search for specific companies in the search bar at the top-left and

explore all available data for that company, like historic prices, multiples, peers, financials, and news. One can also use a screening tool to find a list of companies fulfilling an arbitrary list of criteria. As an example, one can automatically find all stocks with a certain revenue, in a certain sector in a certain set of companies.





To get, for example, stock data on a specific company, one can navigate to the company's chart page and export the data. This is very tedious, though, when data for several companies is desired. The most common solution to this issue is to use the Excel plugin. This plugin allows one to define the tickers and fields one wants data for, and copy-paste formulas in a range to get data for all tickers and all fields. As an example, one can have a list of company tickers vertically in one column and a list of datapoints one wants for those companies (revenue, EBIT, etc.) horizontally in a row. Then, with the help of the formula-builder, one can create a formula that references the tickers and data item names in the rows and columns and gets that data.

Insert a screenshot from the Excel plugin here.

The Eikon Data Item Browser

The most important part of the Eikon data platform to work with in conjunction with the API is the Data Item Browser. The following is adapted from Fosse (2021):

The Data item browser is a tool to browse data items (the term data item is used by Eikon for a type of information regarding an instrument). To find the data item browser, search for it in the top left search field. If entering an instrument, you can see what all available data items for it are. This is useful to explore what data is available for that instrument and how the data is represented. Then use the Formula builder to extract that data item using its Data Item Code.

Tip: To more easily find what you are searching for use the options in the right corner. Displaying blank values can be useful as it may not be blank for other relevant instruments. In the bottom you can sort for relevance and A-Z. In my experience Eikon may not share your perception of relevance, so A-Z can be useful.

The screenshot shows the Eikon ASA Data Item Browser interface. The browser displays a table of data items for EQNR.OL. The table has columns for Data Item Name, Data Item Code, and EQNR.OL. The first row shows '% Difference from 10y Hig...' with Data Item Code 'TR.DiffFromHighBookValu...' and a value of -19.9771. The settings gear icon in the top right corner is circled in red, and a settings menu is open, showing options for DATA ITEM FREQUENCY (Real Time, Static, Time Series), COLUMN DISPLAY (Data Item Name, Data Item Code, Values), and ROW DISPLAY (Blank Values).

Data Item Name	Data Item Code	EQNR.OL
% Difference from 10y Hig...	TR.DiffFromHighBookValu...	-19.9771
% Difference from 10y Hig...	TR.DiffFromHighPE10Yr	-70.1810
% Difference from 10y Lo...	TR.DiffFromLowBookValue...	128.0245
% Difference from 10y Lo...	TR.DiffFromLowPE10Yr	66.8468
% Difference from 15y Hig...	TR.DiffFromHighBookValu...	-45.3246
% Difference from 15y Hig...	TR.DiffFromHighPE15Yr	-70.1810
% Difference from 15y Lo...	TR.DiffFromLowBookValue...	128.0245
% Difference from 15y Lo...	TR.DiffFromLowPE15Yr	102.8234
% Difference from 2y High...	TR.DiffFromHighBookValu...	-12.4547
% Difference from 2y High...	TR.DiffFromHighPE2Yr	-70.1810

Why should we use the Eikon API?

Some basic information on the official Eikon Python package we base this project on can be found [here](#). More in-depth documentation of the API is in [this pdf](#).

One important reason for using the API instead of the Excel plugin is because important limitations of Excel/Eikon:

- Eikon only allows fetching of data for 7 500 tickers (companies, bonds, commodities, etc.) at a time—getting more means creating several Excel sheets and merging them manually.
- Excel only accommodates a little over one million rows and 16 000 columns—very little in our modern big-data age—and if more is required, well tough luck solving it with Excel.
- Excel is slow and inflexible when it comes to handling and manipulating data, and when fetching data from Eikon's servers, especially when the number of rows grows.
- Excel is not particularly well suited for fetching and working with time-series data.
- Fetching data with Excel requires a lot of manual manipulation of the rows, columns, and formulas to get data—Python is much more flexible and automatable.

But Eikon already has a Python API, why does this project exist?

It is true that Eikon has a python package that allows one to get data with Python. However, they have not made it easy for people to use—rather the opposite—at times it seems like they have tried to make users' lives as hard as possible. This project, then, generalizes much of the code we painstakingly arrived at through countless hours of debugging Eikon errors. You can of course ask, "What's the point if it's so hard?" We agree, but if you want large amounts of data current financial data, there is no other source available to us than Eikon (as far as we know), and Excel will just not cut it.

Some issues to be aware of in the vanilla Eikon API

- There are two main methods to get time-series data: `get_data` and `get_timeseries`. When each shall be used is not intuitive.

- Eikon has a tendency to time out or throw some kind of error when requesting a lot of data. This can be due to the amount of data requested in a single request or the number of requests in a day
- When using `ek.get_timeseries`, the available fields vary with what type of data you request. When requesting stock quotes, "CLOSE" and "OPEN" are available, while "VALUE" is not. When requesting macro series, the "VALUE" field is populated while all the others are not.

How we have addressed the issues

We have tried to address the main pain points we experienced in the making of our Master's. This involves code that simplifies the dichotomy between the two data functions, code to handle batching and waiting when getting large amounts of data overnight, and handling of common errors.

Other Eikon

- Finding fields: Select a small set of tickers and all data items that might be of interest. Run `get_data` without specifying a filename and inspect the resulting dataframe to see which data items are valid for this kind of entity.

Overview of Pytho package

Examples

Equity

Bonds

Macro

Sources

Fosse, Henrik Giske (2021) "The complete guide to extracting bond data from Eikon (as far as I know)"