**Why and how I built a UK Energy dashboard from scratch?**

Brought together many skills I’ve been learning:

* html, css and javascript for front end
* python for backend
* using an API
* parsing data
* data analysis
* potential for use with machine learning

Using skills and knowledge I already have from my previous job as an energy trader

* I know where to get the data from
* What the data means and how to use it

It’s an industry of interest to me, therefore the project was exciting and good way to keep interested in it – this is especially important when it gets tough and the code just won’t work (and this will happen at some point)

It’s applicable to my next venture in Japan, setting up an energy trading desk. These are the exact kind of skills I need to become familiar with and apply to a business case

Learned:

* how to host a simple server
* pulling and parsing data using API
* how JavaScript and JQuery work together (using ajax)
* how to pull data in from a server and display it
* start reading the console errors
  + e.g. couldn’t get some of the charts to work, realised it was a highcharts error #16 – I was importing both <https://code.highcharts.com/highcharts.js> and <https://code.highcharts.com/stock/highstock.js> which caused problems. Only needed the latter if using charts and stocks

Issues:

* parsing with Beautiful soup
* getting highcharts to display data when using getElementById
  + turns out solution to this was order of importing sources, needed to have highcharts.js before modules/data.js
* sorting through a complicated JSON/dictionary data
  + using simplejson to help better decode error (ValueError: No JSON object could be decoded) to (simplejson.errors.JSONDecodeError: Expecting value: line 1 column 1 (char 0))
* terrible openweathermap data, solution found but they also remove the keys from the data if there is no value so have to include exception rules as well
* difficult x axis with date-time
  + sorted this for a few graphs, became more troublesome with forecast generation and demand (snapshot), because I tried to divide the number of weeks by 4 and do some weird maths. This worked to an extent but it doesn’t divide equally so gave the weird floating columns
  + used moment.js library to solve

Draft

This article will be the first I’ve written on the topic of programming. During my very short ‘career’ in programming (and by career I mean learning not actual pay as of yet) I have read many articles myself, whether it be a list of the best courses to take, websites to learn from, books to read, top 10s for junior developers, roadmaps for aspiring data scientists… etc. I’m sure you’ve seen these articles everywhere, their numbers grow by the day and I personally find it quite daunting to keep up to date with. Especially as someone who is self learning such a huge area of study, not within the structure of a university degree.

I’ve decided the best thing for me to do now is to consolidate my learning into something more tangible. This could be in the form of a personal project, an article or in this case, both. One thing I have taken away from the hundreds of articles and blogs I’ve now read is that my efforts need to be visible to others in the form of a portfolio. I have taken steps to create this portfolio over the past year almost, mostly in the form of my own website. This recent project is a bit different and is, in my opinion, an excellent way of putting together my new skills and pushing myself that bit further (more than I expected I came to learn). I also decided to write this article about it:

1. to add a story and context to a bunch of code
2. to draw attention to issues I had and their solutions for others to learn
3. to receive feedback from more knowledgeable programmers for improvements

My project of choice was to design a dashboard for the UK energy market that could be viewed in a web browser and updated live. I chose this project for several reasons; **my past experience and knowledge, my current languages of choice being Python and Javascript, the use of external APIs to retrieve data, both front-end and back-end aspects, and data analysis.**

**Past experience and knowledge**

I’m starting the writing up of this article during a particularly difficult part of the project. One of those issues when the data is horribly structured and refuses to play nicely – it’s driven me crazy. Beginning to write this article has given me fresh motivation and reminded me of the reasons for starting the project in the first place. Which brings me to my first point. It’s an industry of interest to me, therefore the project is exciting and a good way to keep interested in it – this is especially important when it gets tough and the code just won’t work (and this will happen at some point). I am using skills and knowledge I already have from my previous job as an energy trader, meaning I know where to get the data from and what the data means and how to use it. It’s also applicable to my next venture in Tokyo, setting up an energy trading desk. These are the exact kind of skills I need to become familiar with and apply to a business case.

**Full-stack project**

I’m currently focusing on becoming an effective user of Python and JavaScript. I’ve opted for using Python for back-end data fetching, data manipulation and analysis. I’m also using the standard three pillars of web development for the front-end; HTML, CSS and JavaScript. Alongside these come several frameworks and libraries such as numpy, pandas, JSON, Ajax and others that I decided to use as I went along. One of the biggest learnings I wanted from this project was how these languages interact with one another – something you don’t really learn from doing bucket loads of problems on codecademy, freeCodeCamp and Code Wars. Not to dismiss how instrumental all these resources have been to my understanding of computer programming and problem solving using code.

**Interacting with external sites**

I have very little experience using and implementing an API, though I have known for some time what they are used for. I decided it was finally time to put this to use in my own project of choice, become familiar with the different types of responses that can come from various websites and the challenges that come with handling these different data structures.

**Potential for further work**

Alongside basic programming and computer science reading I have been doing over the past months, I have also been studying machine learning. I am most of the way through the Machine Learning Engineering course on Udacity now and will be seeking to continue my learning through my own projects. This project will have already done most of the groundwork in terms of data collection and cleanup – which can be a huge part of the machine learning process. It only seems logical to put the collected data to further use.

So I’ve pretty much covered why I decided to take on this project, now I’m going to delve into how I went about it, the issues I faced and what I learned.

**Project Plan**

First things first, I laid out a rough plan of what I wanted to achieve and how I would go about doing it. This is an integral part of any project in and outside of work and can be overlooked when working on something alone I think. There’s no need for this to be an exact replica of what the end product will look like, but it gives a good indication of where the project should be heading and is a good reference point to keep a clear picture of your goal when you get too bogged down with in-depth coding. To this end, I ‘sketched’ up a quick layout of the dashboard, created a list of the different information that would be shown and drew out a rough architecture of how each part would be built, using which languages and how they might interact. Being my first independent programming project by no means would this be perfect but this improves with practice and understanding.

**Dashboard design**

This is a very scrappy outlook of what I had planned for the dashboard. It wasn’t really intended for anyone’s eyes, only for me to have a picture of what my target was. I’d agree with anyone that says designing a graphic in excel is verging on blasphemy.

**Data collection**

I found this step to be relatively straightforward. Once I had got my head around the nuances of the individual APIs it was quite an enjoyable and satisfying experience of specifying some inputs then see my console fill up with a stream of data. Data that meant nothing to me at this point. The sites that I pulled data from were Elexon (mostly), OpenWeatherMap and …, so each case had subtle differences. But for the most part I used the same code, there was no need to completely work from scratch every time, and I created formulas to iterate over different data pulls where possible.

Used httplib2

**Data cleaning**

This was arguably the most frustrating step for so many reasons. I’m sure anyone who has parsed either xml or JSON data will tell you the same thing. Rather than complain about all the different issues I faced here I will summarise how I got through it.

Firstly I used Beautiful Soup, a well known Python import, to parse the Elexon data.

* sorting through a complicated JSON/dictionary data
  + using simplejson to help better decode error (ValueError: No JSON object could be decoded) to (simplejson.errors.JSONDecodeError: Expecting value: line 1 column 1 (char 0))

**Backend to frontend**

**Auto updating**

When it came to updating the data on the dashboard I came to a bit of a standstill. The idea I had was to create a button on the dashboard that would make a call to the main.py python script and pull in the latest data. This is when I realised that this was a lot easier said than done. Instead I came up with a more elegant solution to use launchd in macOS to run the fetch functions on a weekly/daily/hourly basis. This had the added benefit of keeping my dataset up to date without any extra work from me and I learned about this tool that will no doubt be useful for future projects.

* permission issues
* plutil to check file and console.app

Though I haven’t completely forgotten about the issue of knowing how to call a python script through JavaScript, which would be useful to know. I plan to do some research into Flask, a python web framework that looks like it should do the job. If there are any suggestions for this please let me know!