

You have **2** free member-only stories left this month.

[Sign up for Medium and get an extra one](#)

Jason

Follow

Nov 8, 2022 · 4 min read · ·

Save

How a Simple Script Helped Make Me over \$1000/month

I'll be going over how I did it and how you, too, can apply it today

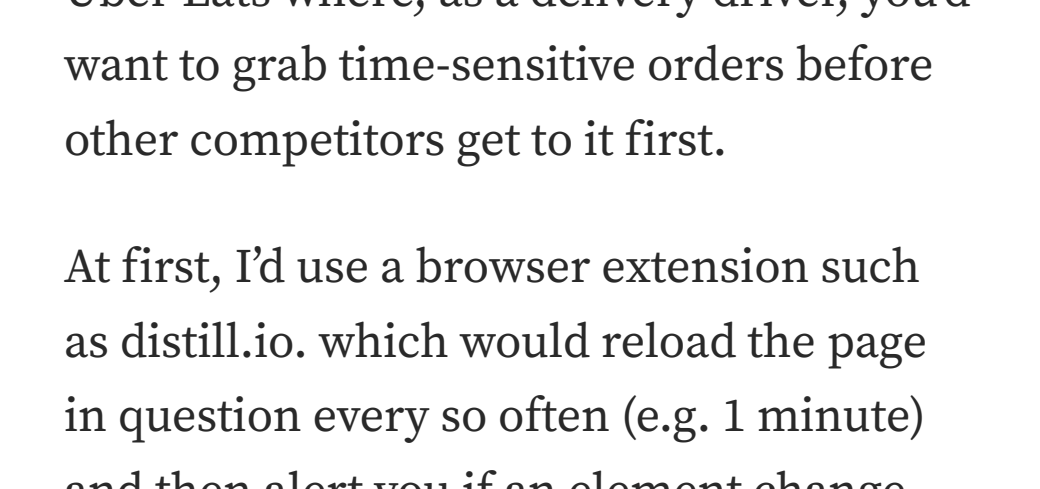


Photo by [Caspar Camille Rubin](#) on [Unsplash](#)

Yes, you heard that correctly — by writing a single-file script and simply leaving it running, I was able to automate a workflow process that was crucial to making money for me without constantly having this looming thought in my head.

Disclaimer: This is by no means 100% passive and may not work for everyone, but hopefully going over my thought process could help inspire you to build your side project/script with a little more incentive (money!)

Motivation

Like a lot of others during these tough economic times, I, too, started looking for ways to earn some extra side income. I'm sure many will not be foreign to platforms that are part of this gig economy, such as Uber Eats where, as a delivery driver, you'd want to grab time-sensitive orders before other competitors get to it first.

At first, I'd use a browser extension such as [distill.io](#). which would reload the page in question every so often (e.g. 1 minute) and then alert you if an element change occurs (i.e. there's new data => order = money):

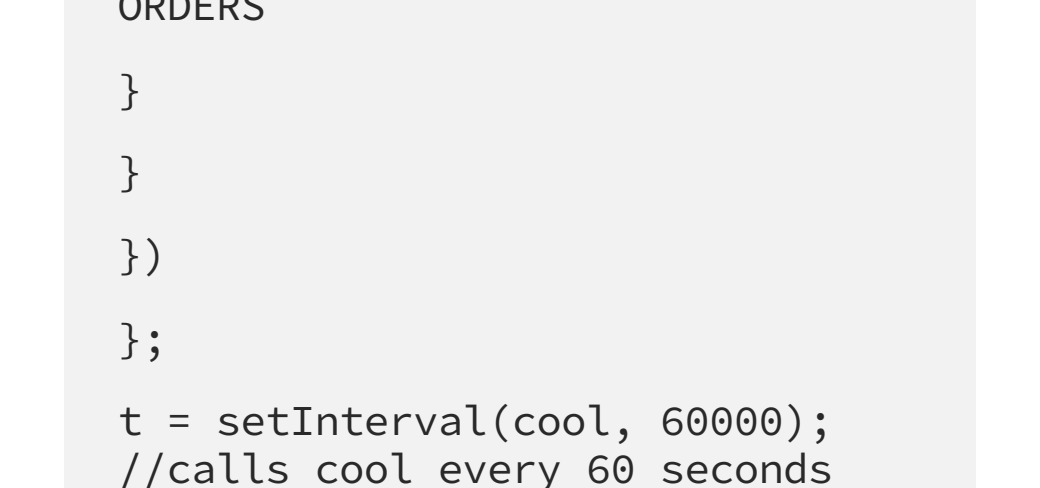


Here, [distill.io](#) allows you to select an element on a page and check for changes per refresh

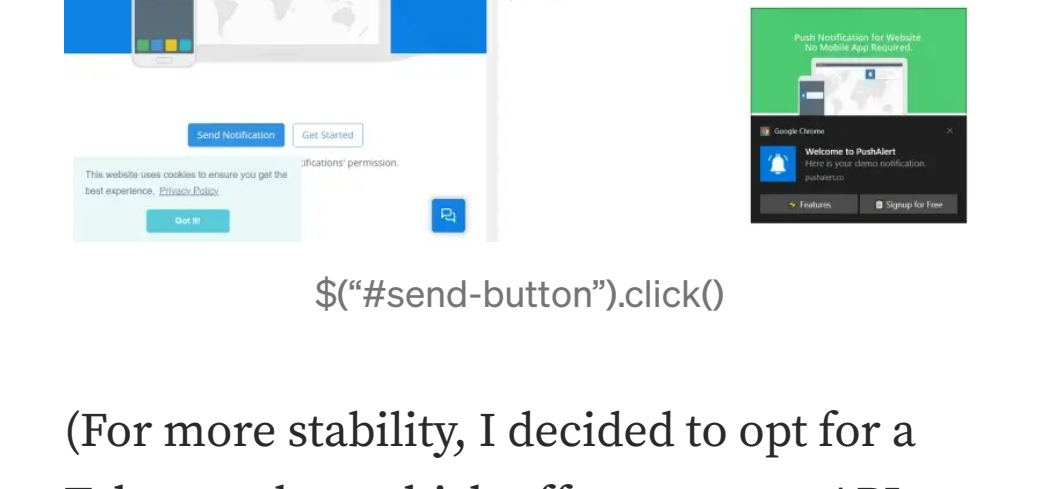
Why Use Code Then?

Eventually, I realized that sometimes, it's not enough to just simply for alert to pop up, then frantically try to click back to the page and manually accept the order.

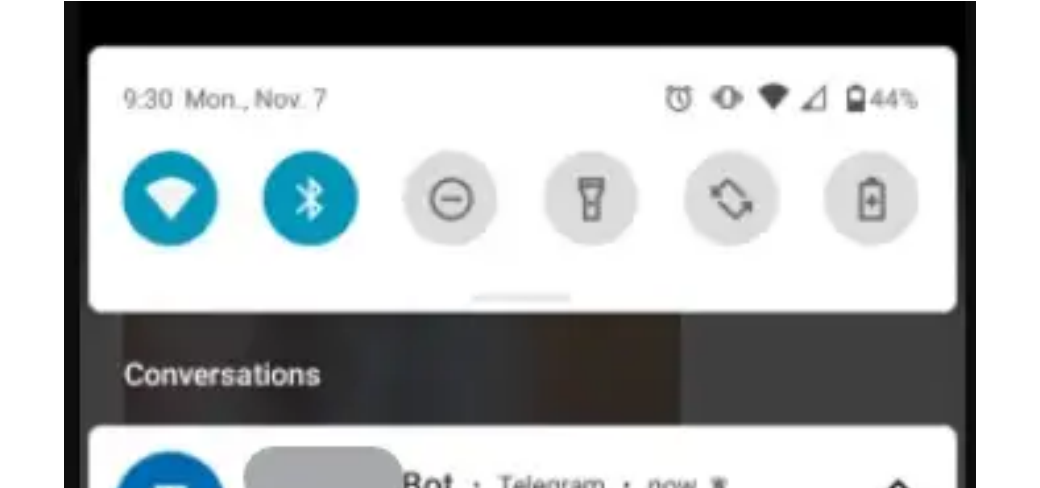
To assume more control, I decided to sniff out the API call behind-the-scenes that achieves this, which can be done by first opening DevTools:



Now, click on the Network tab, refresh the page, find the API call that you're looking for (in this case, it's `getAvailableJobs` as the name implies, else refer to documentation, if any), and copy as fetch:



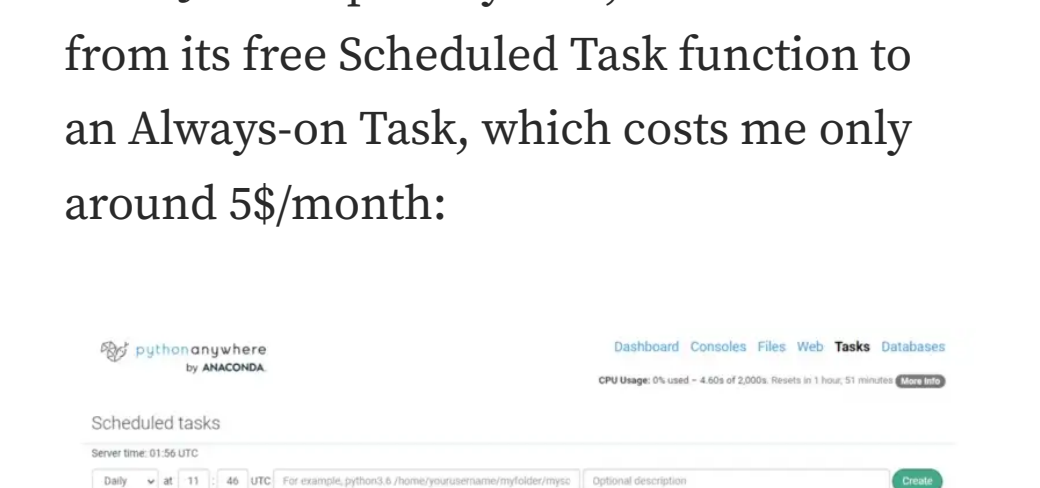
You can simply test out the API call directly in the Console as follows:



Now wrap the call with the following template:

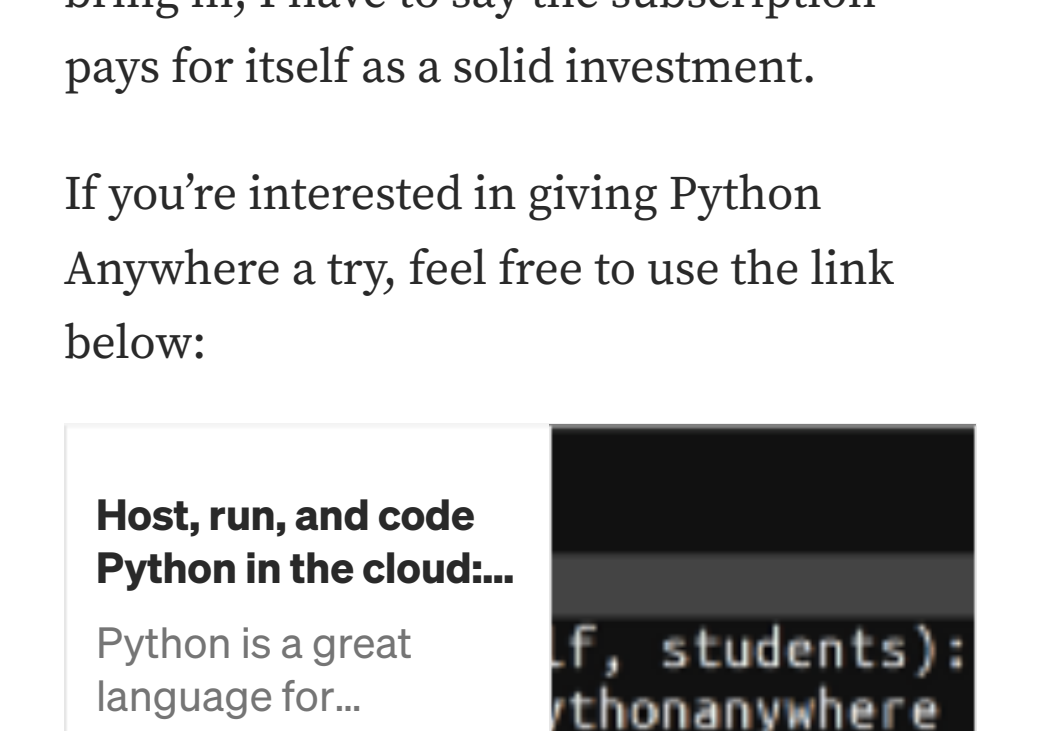
```
cool = function() {  
  //INSERT ABOVE FETH CODE HERE  
  .then(response => response.json()).then(data => {  
    if ({CONDITION IS MET (e.g. delivery time is reasonable)}){  
      fetch("https://[YOUR_API_HERE]", {  
        //USING THE SAME INTUITION ABOVE, FIND API CALL TO ACCEPT ORDERS  
      })  
    }  
  })  
};  
t = setInterval(cool, 60000);  
//calls cool every 60 seconds
```

For push notifications, I used to just run the above script on the Console while being on a website with a push alert button and borrowed it:



`$("#send-button").click()`

(For more stability, I decided to opt for a Telegram bot, which offers a power API to send push notifications to any of your devices with Telegram — might do another tutorial on that if I have enough time)

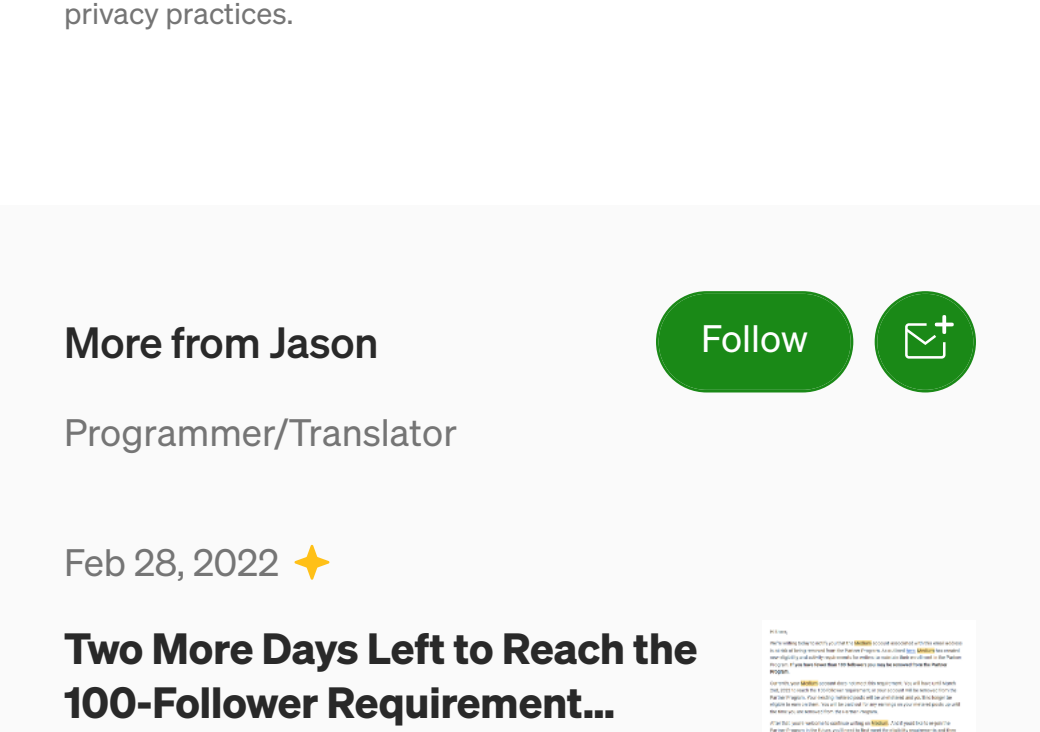


A very simple way to set up push notification for FREE

Where Should I Run This?

Of course, it can become a hassle to try to set it up every time. Other than leaving it run on my computer 24/7, I was scratching my head looking for the best way to host this. I started using an AWS EC2 Instance, which was a pain to set up and would still require me to check the VM terminal every so often to see if my script is still running.

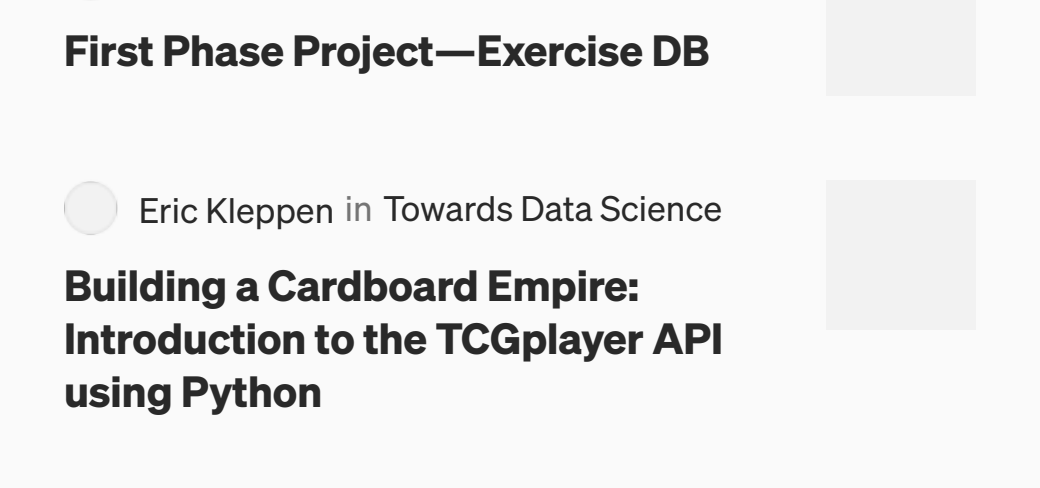
Since it was CRUCIAL that my script stays live 24/7 (a matter of \$\$ vs. no \$\$ to be gained), I needed to make sure that the script would restart if say a crash were to happen. That's when I discovered Python Anywhere and, after converting my code from Javascript to Python, soon switched from its free Scheduled Task function to an Always-on Task, which costs me only around 5\$/month:



Scheduled task run periodically, while always-on run and restart 24/7

And trust me, I'm not the type of person to pay for services, but for the peace of mind I get in return and the money it helps bring in, I have to say the subscription pays for itself as a solid investment.

If you're interested in giving Python Anywhere a try, feel free to use the link below:



And that's all for now! Sorry if I couldn't get into all the nitty gritty and missed anything — some tweaks, etc. may be necessary depending on the nature of the website you're using. Please leave a comment/contact me if you need any more specific help, thanks.

Make Money Online Programming Coding Automation Lifehacks

1.5K 28

Enjoy the read? Reward the writer. [Beta](#)

Your tip will go to Jason through a third-party platform of their choice, letting them know you appreciate their story.

[Give a tip](#)

Get an email whenever Jason publishes.

Your email

[Subscribe](#)

By signing up, you will create a Medium account if you don't already have one. Review our [Privacy Policy](#) for more information about our privacy practices.

More from Jason

Programmer/Translator

Feb 28, 2022

Two More Days Left to Reach the 100-Follower Requirement...

Follow 2 min read

Share your ideas with millions of readers.

[Write on Medium](#)

Nov 4, 2019

日本語の独学からネット翻訳者へ

日本語 3 min read

Recommended from Medium

Karishma Babu

Building responsive pages on Bubble

Lam Duong

First Phase Project—Exercise DB

Eric Kleppen in Towards Data Science

Building a Cardboard Empire: Introduction to the TCGplayer API using Python

Dylan Reuter

Hack the Box: Jail Write-up

Steven Yue

An Investigation of My Home Thermostat

hazirafidi

How to Install and Integrate Spark in Jupyter Notebook (Linux or WSL)

Patrick Kelly

Stupid simple go code “generation”

Elizabeth B White

[https://t.co/rISOWUfuxR](#)