PythonAnywhere Lab and building a Time API in Flask

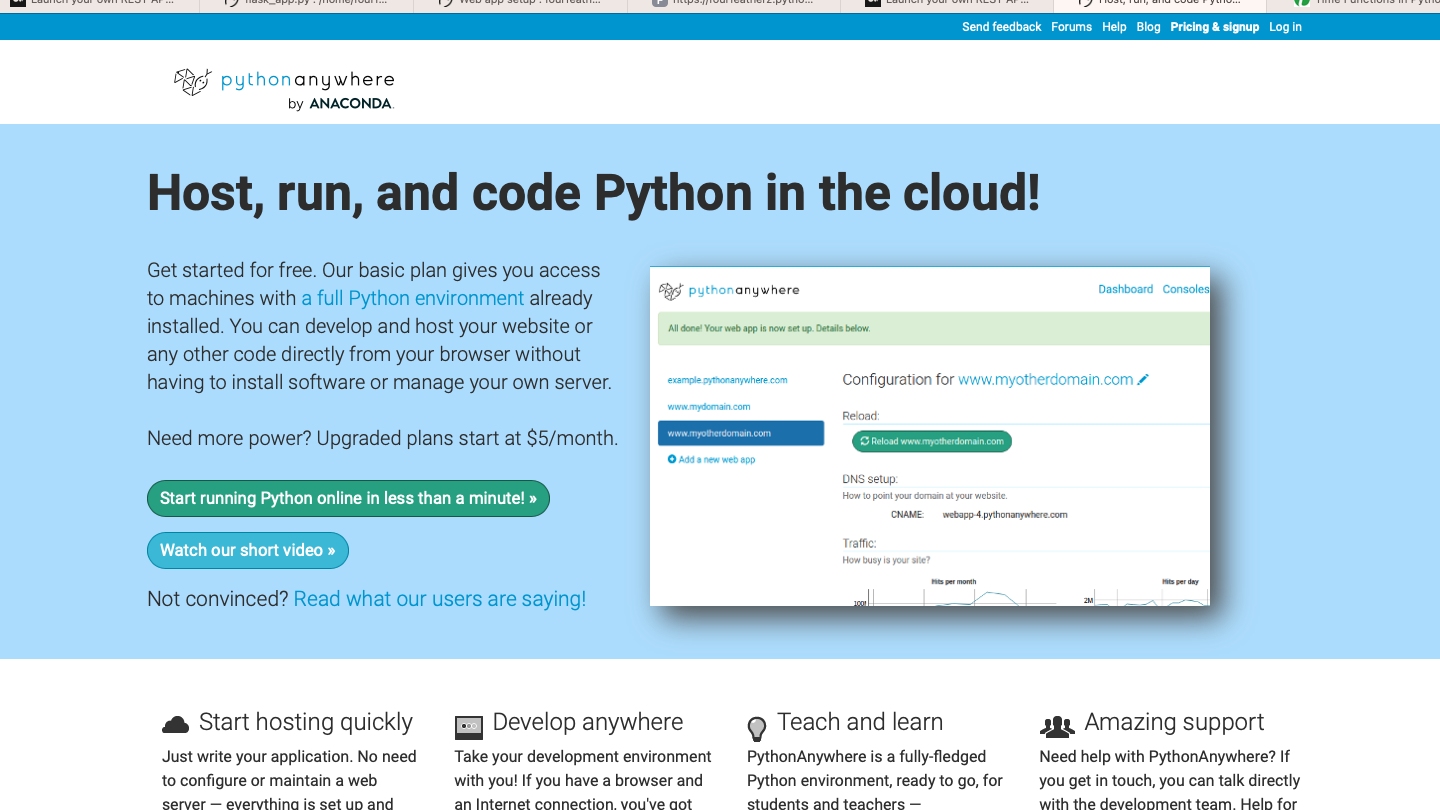
This lab will be important for the end of the semester project when you are building your API based chatbot, which will speak to Discord. In today’s lab, we’ll apply the ideas of the REST API but calling GET functions. You’ll be building this basic API server on PythonAnywhere, which is a free service for what we need.

The steps will be:

1. Open a Python Anywhere Account
2. Build a Flask Web App Shell
3. Create your customer API Endpoint
4. Test your results

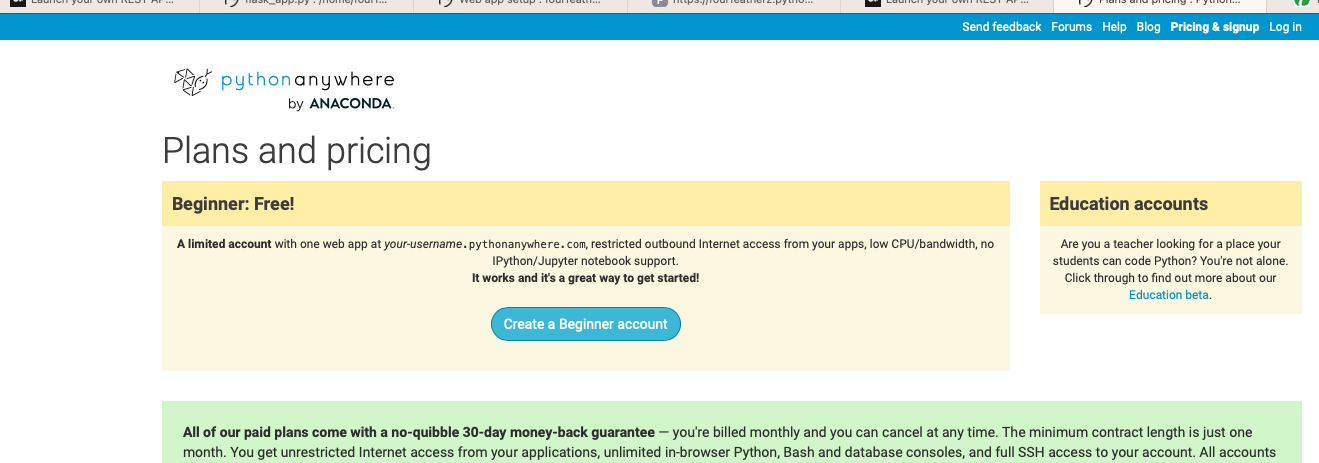
STEP 1

Open your account: Navigate to PythonAnywhere and click the top right of Pricing and Signup

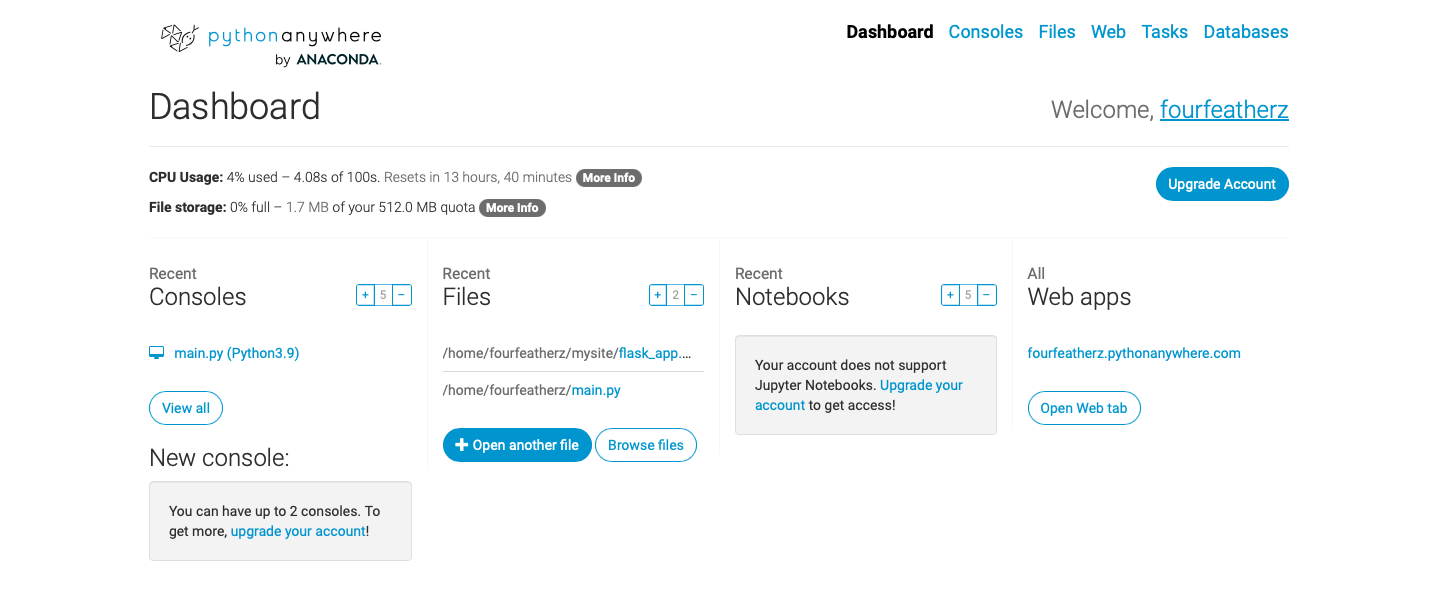




Next you will select “Create a beginner account”



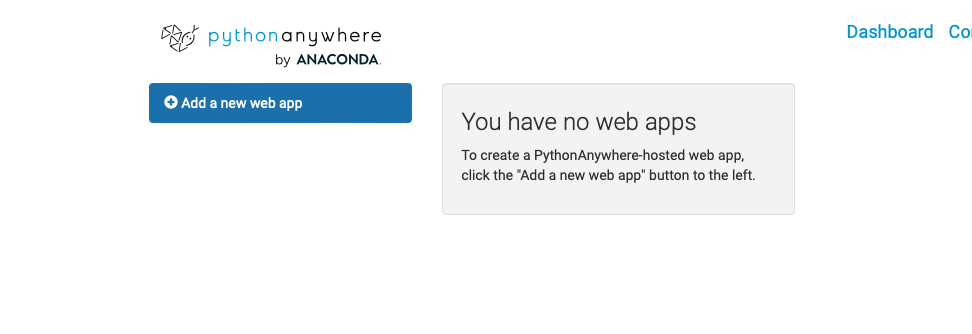
This will bring you to your blank Dashboard.



Here you can do a lot of things for free. You can run Python at the console, build notebooks or host Web Apps. That is what we will do. We will use Flask, which is a python framework for delivering web applications. It will manage the routing, error codes and the HTML server for us.

STEP 2

So, you’ll click on the “Open Web Tab” under Web Apps on the right and you should see a screen like the following:

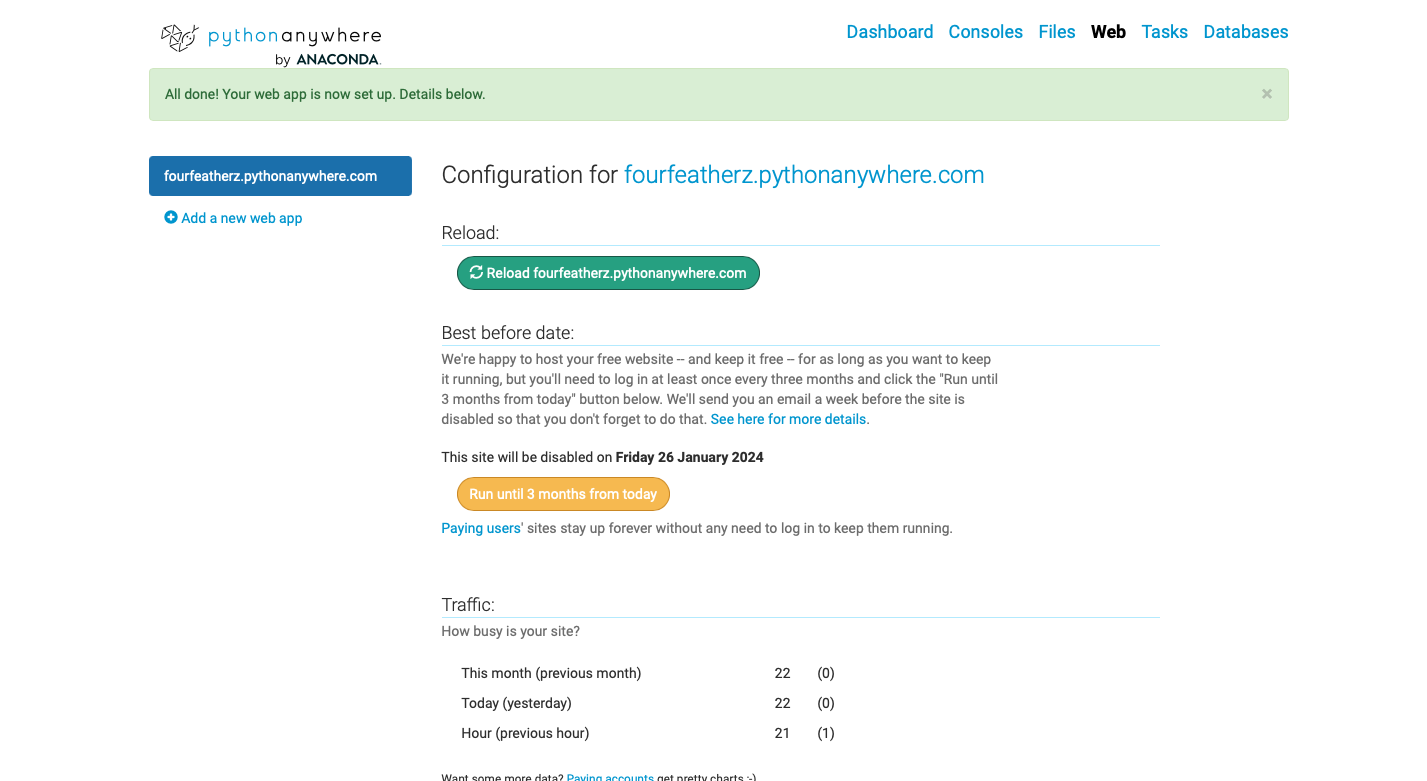


You’ll go ahead and “Add a new web app” and click next.

Next you’ll see a prompt to select a Python Framework. Click “FLASK” and then choose python 3.9, the most recent version.

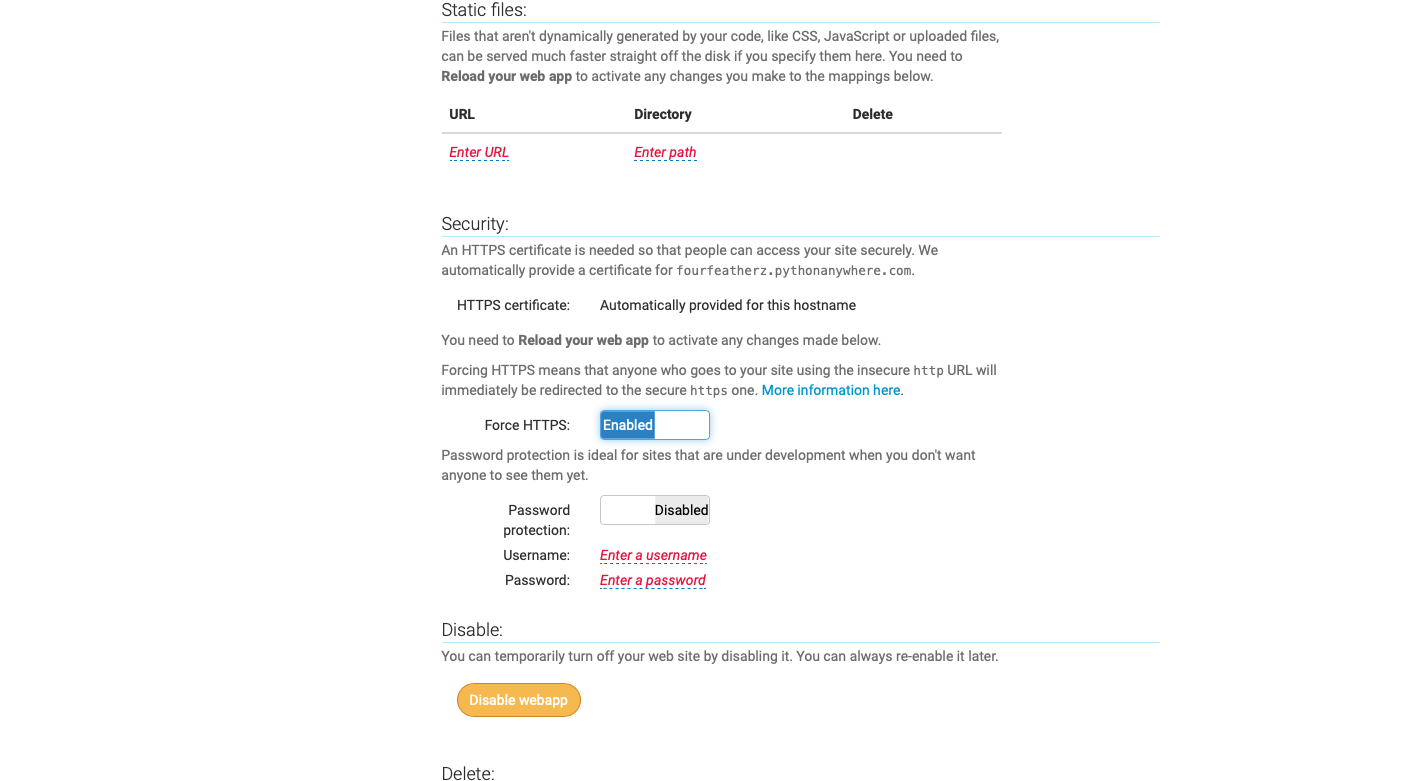
The default path is fine…click next to create.

You’ll now see the basic configure screen…we won’t need to do much here.



If you click on the link after “Configuration for….” (in my case fourfeatherz…) it will load your web site.

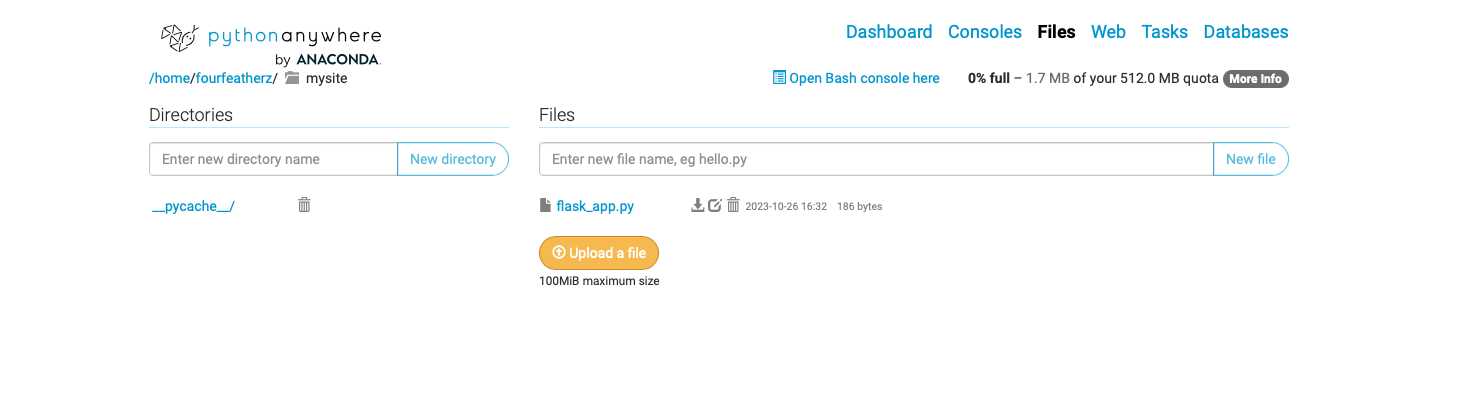
Now we need to make one adjustment and we can get to coding….scroll down and make sure “Force HTTPS” is enabled.



STEP 3

Now lets look at our base code. Scroll up a bit and click on “Source code – Go to Directory”

It should bring you here where you see flask\_app.py. That is clickable…so open it by clicking.



You’ll see your basic method that creates 'Hello from Flask!'.   
# A very simple Flask Hello World app for you to get started with...

from flask import Flask

app = Flask(\_\_name\_\_)

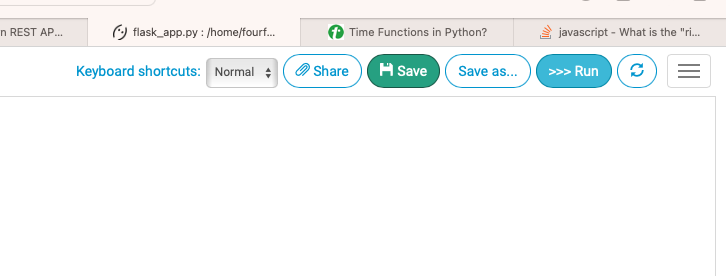
@app.route('/')

def hello\_world():

return 'Hello from Flask!'

Here the application is listening for just the root website…and when it finds that, it will execute Hello\_World(). Go ahead an experiment with the return method and see how it changes the website response. You do that by hitting save in the top right corner and the click the Refresh Circle to restart the server, after editing your code.







**Step 4**

Now we need to make our own customer endpoint.

Here we will need to alter the code to look for a new route. Let’s create an endpoint called “timeofday” with a REST method of GET

Add this code above the existing app.route

@app.route('/api/v1/timeofday/', methods=['GET'])

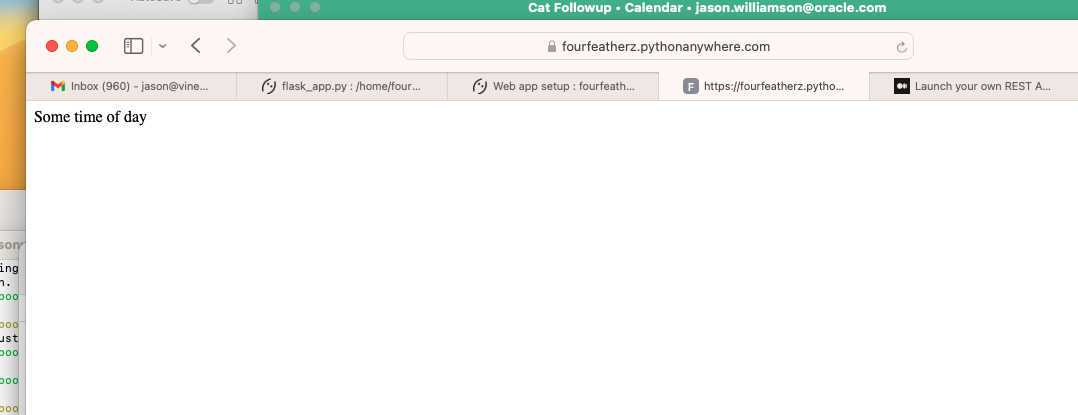
def timeofday():

return “some time of day”

Here we are saying that this is an API and version 1. We aren’t taking any parameters at this point. Save, Refresh and enter this new endpoint into your url. In my case it’s

<https://fourfeatherz.pythonanywhere.com/api/v1/timeofday/>

Here are my results.



Let’s go ahead and make it real.

Update your code to use the time library.

Add your import the top of the file

**from** datetime **import** datetime

and alter the return statement to automatically return the time on the server. Note we need to convert the DateTime object into a string….you can learn more about different string formats here 🡪 https://www.freecodecamp.org/news/python-datetime-now-how-to-get-todays-date-and-time/

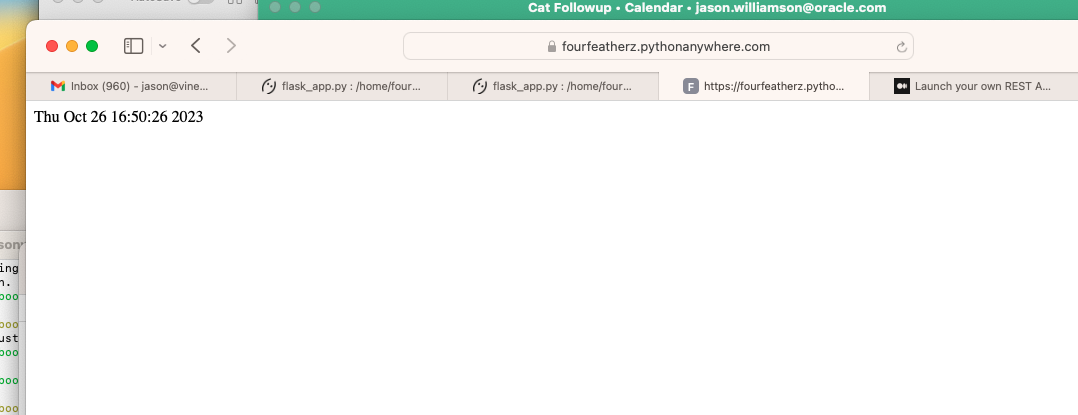
def timeofday():

current\_dateTime = datetime.now()

currentTimeStr = current\_dateTime.strftime("%H:%M:%S")

return (currentTimeStr)

Save – refresh and the reload.



Notice the Time zone…it’s UTC.

Let’s go ahead and put it in East Coast Time. We use the library pytz for that, so import that at the top under your datetime import.

Now alter your code to set the TZ as New York and call that time zone.

newYorkTz = pytz.timezone("America/New\_York")

current\_dateTime = datetime.now(newYorkTz)

currentTimeStr = current\_dateTime.strftime("%H:%M:%S")

return (currentTimeStr)

Now you know how to set a basic API, and we’ll use this to make our chatbot be able to listen to questions and answer them in our next iteration.