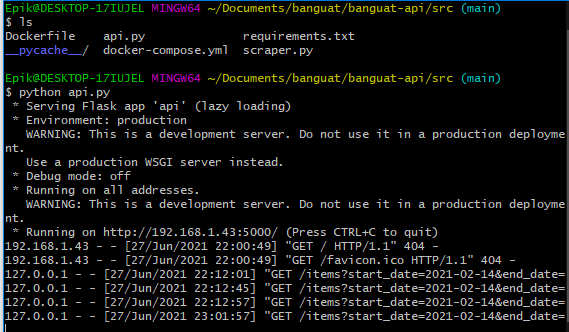
Link enlazado a git hub en mi cuenta el nombre solo lo deje ahí por si no me crea nombre infantil

<https://github.com/SkayBlack/banguatAdder.git>

Primero el levantado del servidor con el siguiente comando ya que primero se tiene que instalar Python ejecutar el archivo api.py con el comando ( Python apy.py) pero primero instalando unas cuantas librerías para que este funcione correctamente que por la cual son

* pandas
* lxml
* flask
* flask\_restful

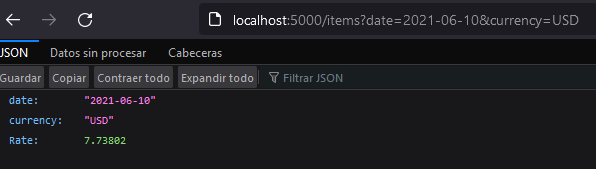
y seguidamente el comando especificado anteriormente.



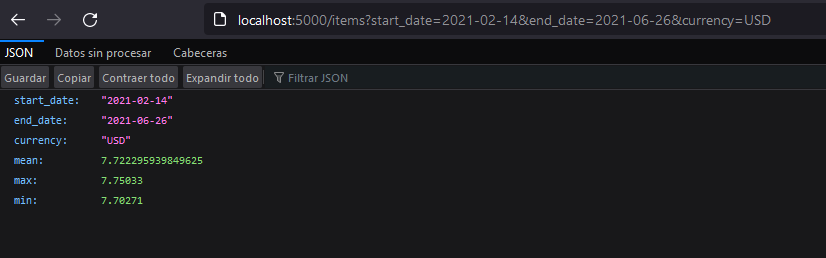
Pruebas de la consulta de Banguat en donde me retorna el valor del Dólar en Quetzales que por la cual no solamente podemos solo en una fecha en específico, sino que también podemos ver en un rango de fechas en la que se puede observar cual a sido el valor del dólar a Quetzales ya que por defecto acá esta como un solo dólar solo se tendría que cambiar el valor de cuantos dólares queremos cambiar

Link en la que se ejecuta para poder visualizar cual es el valor del dólar en esa fecha en especifico

<http://localhost:5000/items?date=2021-06-10&currency=USD>



Podemos cambiarlo y colocar la fecha en especifico

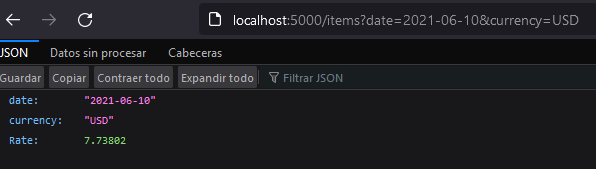
En rangos de fecha con este link directamente <http://localhost:5000/items?start_date=2021-02-14&end_date=2021-02-17&currency=USD> aparecerá los valores

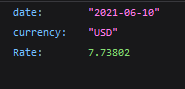
Fecha Fin

Fecha inicio

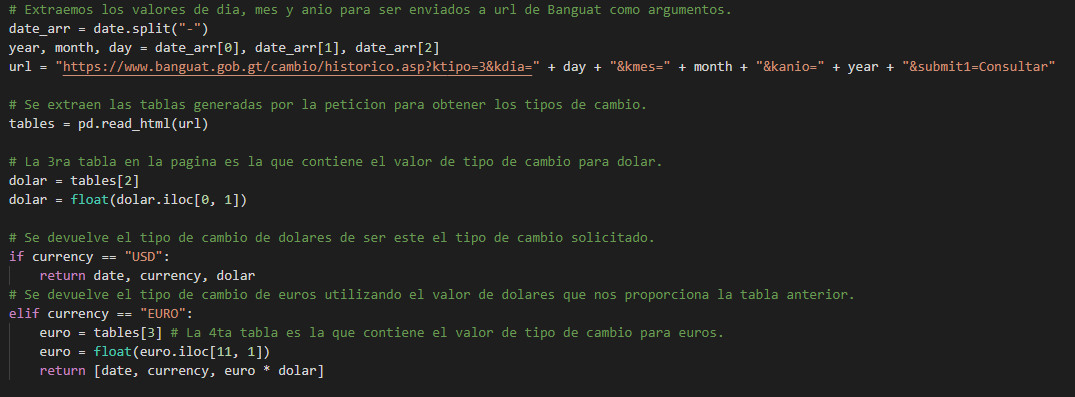
Podemos cambiarlo y colocar la fecha en específico de principio a fin

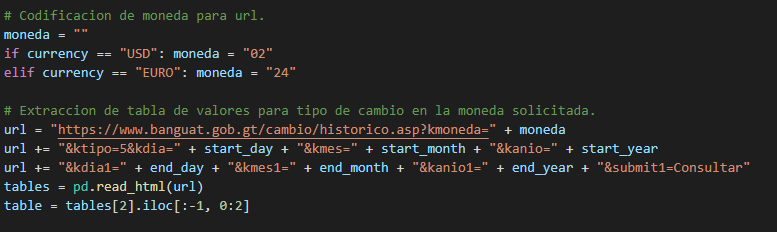
En donde podemos ver un link con una fecha que por la cual nos retornara el valor como predeterminado ahora tiene como dólar



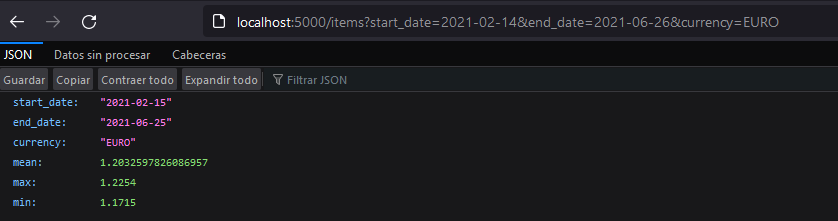
Acá podemos ver que se hizo un consulta a través de la fecha así obtendríamos al tanto el valor del dólar al día cuando nos retorna en el JSON

Se hizo una consulta en la fecha 2021-06-10 y el valor en ese día el dólar es de 7.73 dando a conocer que si funciona ya que el Json ya solo es te multiplicar la variable con el valor retornado

pruebas de que se estableció la conexión con el Banguat

acá podemos ver para la extracción de la tabla ya sea modificando el link para poder ver qué tipo de moneda queremos en este caso se ve que solo hay Euro y Dolar

Valor del euro cambio de link convirtiendo cual es el valor



Partes en la que se desarrolló del proyecto scraper.py

import pandas as pd

import datetime

# Funcion del Metodo 1, recibe la fecha y la moneda y devuelve el tipo de cambio en quetzales para esta.

def single\_date(date, currency):

    # Extraemos los valores de dia, mes y anio para ser enviados a url de Banguat como argumentos.

    date\_arr = date.split("-")

    year, month, day = date\_arr[0], date\_arr[1], date\_arr[2]

    url = "https://www.banguat.gob.gt/cambio/historico.asp?ktipo=3&kdia=" + day + "&kmes=" + month + "&kanio=" + year + "&submit1=Consultar"

    # Se extraen las tablas generadas por la peticion para obtener los tipos de cambio.

    tables = pd.read\_html(url)

    # La 3ra tabla en la pagina es la que contiene el valor de tipo de cambio para dolar.

    dolar = tables[2]

    dolar = float(dolar.iloc[0, 1])

    # Se devuelve el tipo de cambio de dolares de ser este el tipo de cambio solicitado.

    if currency == "USD":

        return date, currency, dolar

    # Se devuelve el tipo de cambio de euros utilizando el valor de dolares que nos proporciona la tabla anterior.

    elif currency == "EURO":

        euro = tables[3] # La 4ta tabla es la que contiene el valor de tipo de cambio para euros.

        euro = float(euro.iloc[11, 1])

        return [date, currency, euro \* dolar]

# Funcion del Metodo 2, recibe rango de fechas y devuelve

def range\_dates(start\_date, end\_date, currency):

    # Extraccion de fecha de inicio.

    start\_date\_arr = start\_date.split("-")

    start\_year, start\_month, start\_day = start\_date\_arr[0], start\_date\_arr[1], start\_date\_arr[2]

    # Extraccion de fecha de fin.

    end\_date\_arr = end\_date.split("-")

    end\_year, end\_month, end\_day = end\_date\_arr[0], end\_date\_arr[1], end\_date\_arr[2]

    # Codificacion de moneda para url.

    moneda = ""

    if currency == "USD": moneda = "02"

    elif currency == "EURO": moneda = "24"

    # Extraccion de tabla de valores para tipo de cambio en la moneda solicitada.

    url = "https://www.banguat.gob.gt/cambio/historico.asp?kmoneda=" + moneda

    url += "&ktipo=5&kdia=" + start\_day + "&kmes=" + start\_month + "&kanio=" + start\_year

    url += "&kdia1=" + end\_day + "&kmes1=" + end\_month + "&kanio1=" + end\_year + "&submit1=Consultar"

    tables = pd.read\_html(url)

    table = tables[2].iloc[:-1, 0:2]

    # Se adquieren las fechas de inicio y fin (cuyos datos se encuentran dentro de la tabla).

    start = datetime.datetime.strptime(table.iloc[0, 0], "%d/%m/%Y").strftime("%Y-%m-%d")

    end = datetime.datetime.strptime(table.iloc[-1, 0], "%d/%m/%Y").strftime("%Y-%m-%d")

    # Se calcula el minimo, maximo y promedio de los tipos de cambio disponibles denro de la tabla.

    # Para el caso de tipo de moneda EURO este resultado es devuelto en dolares.

    min\_rate = min(pd.to\_numeric(table.iloc[:, 1]))

    max\_rate = max(pd.to\_numeric(table.iloc[:, 1]))

    mean = pd.to\_numeric(table.iloc[:, 1]).mean()

    # Devolucion de los datos

    return [start, end, currency, mean, max\_rate, min\_rate]

y el Api.py

from flask import Flask

from flask\_restful import Resource, Api, reqparse

import scraper

import json

app = Flask(\_\_name\_\_)

api = Api(app)

# Define parser and request args

parser = reqparse.RequestParser()

parser.add\_argument('date', type=str, required=False)

parser.add\_argument('currency', type=str)

parser.add\_argument('start\_date', type=str, required=False)

parser.add\_argument('end\_date', type=str, required=False)

class Currency(Resource):

    def get(self):

        args = parser.parse\_args()

        args = {k:v for k,v in args.items() if v is not None}

        if len(args) == 2:

            date = args['date']

            currency = args['currency']

            values = scraper.single\_date(date, currency)

            return {

                "date": values[0],

                "currency": values[1],

                "Rate": values[2]

            }

        elif len(args) == 3:

            start\_date = str(args['start\_date'])

            end\_date = str(args['end\_date'])

            currency = str(args['currency'])

            values = scraper.range\_dates(start\_date, end\_date, currency)

            return {

                "start\_date": values[0],

                "end\_date": values[1],

                "currency": values[2],

                "mean": values[3],

                "max": values[4],

                "min": values[5]

            }

api.add\_resource(Currency, '/items')

app.run(host='0.0.0.0', port=5000)

lo Demas son archivos binaries generado por los mismo librerías que por la cual podrá encontralo en el enlace de github

<https://github.com/SkayBlack/banguatAdder.git>