***Bitcoin Price Alert (Using Python)***

Introduction

Using the popular automation website IFTTT, create IFTTT applets,using Coinmarketcap API.

We’re going to create two IFTTT applets:

One for emergency notification when Bitcoin price falls under a certain threshold; and another for regular Telegram updates on the Bitcoin price.

Both will be triggered by our Python app which will consume the data from the Coinmarketcap API.

In our case, the trigger will be a webhook service provided by IFTTT. You can think of webhooks as “user-defined HTTP callbacks".

Our Python app will make an HTTP request to the webhook URL which will trigger an action. Now, this is the fun part—the action could be almost anything

you want. IFTTT offers a multitude of actions(as i mentioned in the desc) like sending an email, updating a Google Spreadsheet and even calling your phone.

To use IFTTT you’ll first need to set up a new account and install their mobile app (if you want to receive phone notifications from your Python app).

Sending a Test IFTTT Notification Now we can move onto the IFTTT side of things. To use IFTTT you’ll first need to set up a new account and install their

mobile app (if you want to receive phone notifications from your Python app). Once you set that up, we’re going to create a new IFTTT applet for testing

purposes.

To see the documentation on how to use the IFTTT webhooks go to this page and click on the “Documentation” button in the top right corner.

The documentation page contains the webhook URL and it looks like this:

https://maker.ifttt.com/trigger/{event}/with/key/{your-IFTTT-key}

You’ll need to substitute the {event} part with whatever name you gave, when you created the applet. The {your-IFTTT-key} part is already populated

with your IFTTT key.

Creating IFTTT Applets

We need to create two new IFTTT applets: one for emergency Bitcoin price notifications and one for regular updates.

Emergency bitcoin price notification applet:

1.Choose the “webhooks” service and select the “Receive a web request” trigger

2.Name the event bitcoin\_price\_emergency

3.For the action select the “Notifications” service and select the “Send a rich notification from the IFTTT app” action

4.Give it a title, like “Bitcoin price emergency!”

5.Set the message to Bitcoin price is at ${{Value1}}. Buy or sell now! (we’ll return to the {{Value1}} part later on)

6.Create the action and finish setting up the applet

....

Regular price updates applet:

1.Again choose the “webhooks” service and select the “Receive a web request” trigger

2.Name the event bitcoin\_price\_update

3.For the action select the “Telegram” service and select the “Send message” action

4.Set the message text to: Latest bitcoin prices: {{Value1}}

5.Create the action and finish with the applet

Note: When creating this applet you will have to authorize the IFTTT Telegram bot.

in the main function. It will consist of a while True loop since we want our app to run forever. In the loop we will call the Coinmarketcap API to get the latest

Bitcoin price and record the current date and time.

Based on the current price we will decide if we want to send an emergency notification. For our regular Telegram updates we will also append the current price

and date to a bitcoin\_history list. Once the list reaches a certain number of items (e.g. 5) we will format the items, send the update to Telegram, and reset the

history for future updates.

[IMPORTANT]

An important thing is to avoid sending out requests too frequently, for two reasons:

-The Coinmarketcap API states that they update the data only once every 5 minutes, so there’s no point in reloading the latest pricing info more

frequently than that.

-If your app sends too many requests to the Coinmarketcap API your IP might get banned or temporarily suspended. That is why we need to “go to sleep”

(stop the execution of the loop) for at least 5 minutes before we get new data.

**Code**

import requests

from datetime import datetime

import time

TICKER\_API\_URL = 'https://blockchain.info/ticker'

IFTTT\_WEBHOOKS\_URL = 'https://maker.ifttt.com/trigger/bitcoin\_price\_update/n6JpNvUIxgXNd0oO9iVrB7Oi5-ovR73XzcT5gSFeD02'

BITCOIN\_PRICE\_THRESHOLD = 10000

# In this function , the API returns a JSON response, we can convert it to a Python object by calling the .json() function on the response.

def get\_latest\_crypto\_price( url='https://blockchain.info/ticker'):

    response = requests.get(url)

    response\_json = response.json()

    # Convert the price to a floating point number

    return float(response\_json['INR']['last'])

#if you print it you will get latest bitcoin price and you can select bitcoin or litecoin.

#print(get\_latest\_crypto\_price())

#in this post\_ifttt\_webhook function payload will be sent to ifttt service and this will insert our desired event and finally it sends a HTTP POST request to the webhook URL.

def post\_ifttt\_webhook(event,value):

  data = {'value1': value}

  ifttt\_event\_url= 'https://maker.ifttt.com/trigger/bitcoin\_price\_update/with/key/n6JpNvUIxgXNd0oO9iVrB7Oi5-ovR73XzcT5gSFeD02'.format(event)

  requests.post(ifttt\_event\_url, json=data)

#It takes the bitcoin\_history as an argument and formats it using some of the basic HTML tags allowed by Telegram, like <br>, <b>, <i>

def format\_bitcoin\_history(bitcoin\_history):

    rows = []

    for bitcoin\_price in bitcoin\_history:

        # Formats the date into a string: '24.02.2018 15:09'

        date = bitcoin\_price['date'].strftime('%d.%m.%Y %H:%M')

        price = bitcoin\_price['price']

        # <b> (bold) tag creates bolded text

        row = '{}: $<b>{}</b>'.format(date, price)

        rows.append(row)

        # Use a <br> (break) tag to create a new line

    return '<br>'.join(rows)

def main():

    bitcoin\_history = []

    while True:

        price = get\_latest\_crypto\_price()

        date = datetime.now()

        bitcoin\_history.append({'date': date, 'price': price})

        # Send an emergency notification

        if int(price) > BITCOIN\_PRICE\_THRESHOLD:

            post\_ifttt\_webhook('bitcoin\_price\_emergency', price)

        # Send a Telegram notification or it can also send notification in your preferred selected event.I have linked this to telegram and phone android sms notification.

        if len(bitcoin\_history) == 5:

            # Once we have 5 items in our bitcoin\_history send an update

            post\_ifttt\_webhook('bitcoin\_price\_update', format\_bitcoin\_history(bitcoin\_history))

            # Reset the history

            bitcoin\_history = []

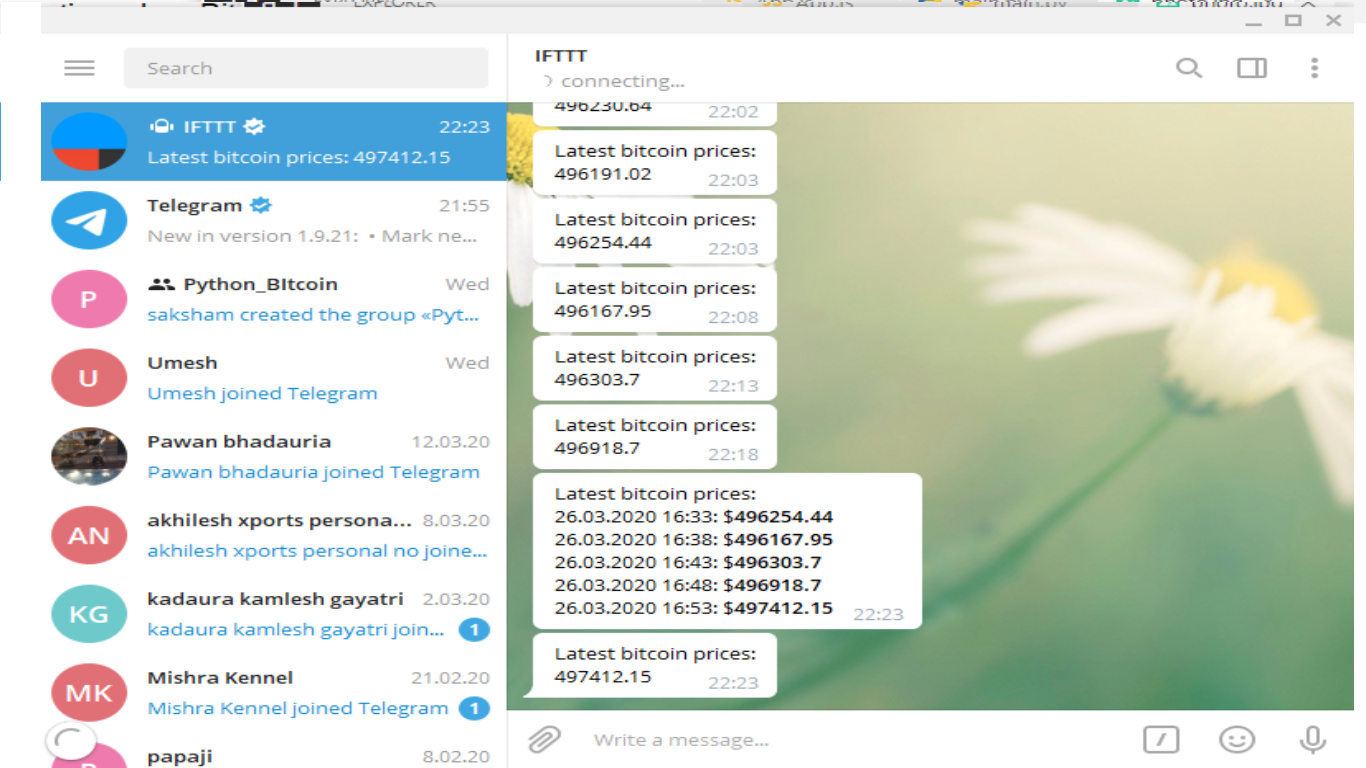
        # Sleep for 5 minutes

        time.sleep(5 \* 60)

if \_\_name\_\_ == '\_\_main\_\_':

    main()

**Screenshot**

****