BITCOIN PRICE TRACKER by Akash Dutta (Robin batch)

Bitcoin, the world‟s most common and well known cryptocurrency, has been increasing in popularity.
It has the same basic structure as it did when created in 2008, but repeat instances of the world
market changing has created a new demand for cryptocurrencies much greater than its initial showing.
By using a cryptocurrency, users are able to exchange value digitally without third party oversight

In this project I made a program to get latest bitcoin price. The general process is to run the
code and then it'll fetch the data from a cryptocurrency API called coinmarketcap and then it'll show
updated price in a public/private telegram channel. For that i have used Python and a third party
service called IFTTT to send the push notification.

#Technology Used:
Python 3.7, IFTTT(for push notification), Telegram(to receive notofication)

#Retrieving the Bitcoin Price:

First, we have to import the requests module and make a function get_latest_bitcoin_price. Now define
the url variable which contains the Coinmarketcap API URL for Bitcoin.
Next, we will populate the 'X-CMC_PRO_API_KEY' with the key we'll get from the coinmarketcap website.
Then we will store the data in a variable response and later will extract the JSON data in 'data' variable
and will return it by dictionary slicing.

#Sending 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 the notification.

To create a new test applet follow these steps:

1. Click on the big “this” button
2. Search for the “webhooks” service and select the “Receive a web request” trigger
3. Let’s name the event 'Bitcoin_Price_Emergency'
4. Now select the big “that” button
5. For the action select the “Notifications” service and select the “Send a rich notification from the
IFTTT app” action.
6. Give it a title, like “Bitcoin price emergency!”
7. Set the message to Bitcoin price is at ${{Value1}}. Buy or sell now! (we’ll return to the {{Value1}}
part later on)

To see the documentation on how to use the IFTTT webhooks go to the main applet 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}
Next up, you’ll need to substitute the {event} part with whatever name you gave our event in step 3, when you created the applet. The {your-IFTTT-key} part is already populated with your IFTTT key. Now will make a variable and will store the webhook url.

Now whenever we call the post_ifttt_webhook function it'll trigger the notification. The event parameter corresponds to whatever event name we gave to our trigger when setting up the IFTTT applet. Also, the IFTTT webhooks allow us to send additional data along with the request as JSON-formatted data.

#Telegram notification update:

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:<br>{{Value1}}
5. Create the action and finish with the applet

To receive the notification in telegram, I created a public channel called BTC_Price and gave the access to the IFTTT telegram bot and make it the administrative section.

Now coming to the python console. We have to create two separate function to get the latest bitcoin price and to update that in our telegram channel. For that in run function we took two separate 'Bitcoin_Price_Update' and 'Bitcoin_Price_Update'. While getting the data from 'get_latest_bitcoin_price' under run function we are returning the value through 'Bitcoin_Price_Update' function which is triggering the Webhook notification applet. And the 'Bitcoin_Price_Update' on the other hand triggering the telegram notification applet. this two applet name and function name needs to be exact same. And lastly the time sleep function imported from datetime package will take care of the current time and the time when we need the data.

The only thing missing is the format_bitcoin_history function. 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>, and so on.

#Command line utility:

For command line utility we imported a package called argparse. Where we added two argument, one for the interval we want and one for the threshold we want. For each we gave two default value which will be called if we failed to give any argument. here two 'add_argument' function are taking the values and at last we are passing both arguments in 'run' function. So when run function is being called we are calling two different function 'Bitcoin_Price_Emergency' for the notification and 'Bitcoin_Price_Update' to update the price through the telegram channel.

That’s it! In little over 50 lines of Python code, I've created your very own Bitcoin notification service.