Alexa API for crypto portfolio

A flask API for making crypto related queries. Supports both user registration and no user registration requests.

technical stack :- - DBMS software: sqlite - API hosted on: pythonanywhere - Flask: python-flask is used to construct the complete API. - Using blockcypher API for fetch crypto balance at address stored in the database - Using BeautifulSoup4 python module to scrape websites for data

How to use

Currently setup endpoint is at https://alexaskill.pythonanywhere.com/
Account independent features :-
1. Price fetch
https://alexaskill.pythonanywhere.com/?action=price&crypto=[CRYPTO] replace [CRYPTO] with btc, eth, doge, ltc, xrp, anything...

```
{
  "change": 4.75,
  "price": 6455.91,
  "price_old": 6451.16
}
```

response will be a JSON with three fields :- 1. change: change in the price in last 24 hours in USD 2. price: current price of crypto in USD 3. price_old: price 24 hours ago in USD

1. Market cap
   https://alexaskill.pythonanywhere.com/?action=cap&crypto=[CRYPTO]
   replace [CRYPTO] with btc, eth, doge, ltc, xrp, anything...

```
{"cap":"$ 112.00 B"}
```

reponse will be a single field:-
1. cap: the total calue of coins in market. Speak this in millions/billions (speak B as billion and M as million)

1. Crypto news
   https://alexaskill.pythonanywhere.com/?action=news

```
{
  "news": [
    "The head of Indian IT trade organization NASSCOM reportedly stated that cryptocurrency is “illegal” in the country, after two men installing a Bitcoin ATM were...",
    -- SNIP ---
  ]
}
```

response will have a single field:
1. news: its an array of strings which are the news headlines top 100 news articles are fetched from: httpss://news.google.com/search?q=cryptocurrency

1. Price prediction (using machine learning)
   https://alexaskill.pythonanywhere.com/?action=predict&date=[DATE]&crypto=[CRYPTO_LIMITED]

   [DATE] can be :-

   1. tommorrow
   2. day_after
   3. DD/MM/YYYY format for anyday until 30 days from now. Server will automatically handle invalid requests and return appropriate response so don't worry about it eg. 21/10/2018 is a valid date but 10/10/2016 is not (past) server will handle it dont worry [CRYPTO_LIMITED] can be :-
   4. btc
   5. eth
   6. doge
   7. ltc

```
{
  "crypto": "doge",
  "days_from_now": 2,
  "prediction": "0.00493"
}
```

response will have the following fields: 1. prediction : predicted price 2. crypto: name of crypto for which the prediction is requested 3. days_from_now: number of days between today and the date prediction is asked for

Account dependent features:-
NOTE: for testing purpose use ALEXA_ID = 1, i have added this to the database, it will get your results. In other cases, replace the alexa ID by the actual alexa ID, if its registered you will get a response, if not you won't get any response in that case say "Sorry you are not registerd with us, please register your alexa id on the link written in description". I am handling the registration process so dont worry about it, for now assume ID=1 is registered with all four addresses in our database.

1. fetch value of porfolio:
   https://alexaskill.pythonanywhere.com/?action=fetch&id=[ALEXA_ID]&crypto=[CRYPTO_LIMITED]
   replace [ALEXA_ID] with alexa_id, dahiya should know how to get it. For testing use id = 1
   replace [CRYPTO_LIMITED] with :-
   1. all: fetch the value of the complete portfolio, all of it

2. btc: fetch the amount of bitcoins held
3. eth: fetch the amount of ether held
4. ltc: same
5. doge: same

```
{
  "balance": 524.0768936513,
  "btc": 0.04449209,
  "btc_USD": 287.0874353295,
  "doge": 0,
  "doge_USD": 0,
  "eth": 1.16594243,
  "eth_USD": 236.98945832180002,
  "id": "1",
  "ltc": 0,
  "ltc_USD": 0
}
```

the reponse will have :-

```
1. balance: total balance in USD
2. x: number of crypto x held
3. x_usd: amount of cryptp held in USD
4. id: alexa ID of user
```

1. unconfirmed transactions
   https://alexaskill.pythonanywhere.com/?action=unconfirmed&id=[ALEXA_ID]
   replace [ALEXA_ID] with alexa_id, dahiya should know how to get it. For testing use id = 1

   ```
   {
   "btc": 0,
   "doge": 0,
   "eth": 0,
   "ltc": 0,
   "total": 0
   }
   ```

   reponse will have the follwing fields :-

   1. btc: no of btc unconfirmed transactions
   2. doge: no of doge...
   3. ltc: no of...
   4. eth: ...
   5. total: total unconfirmed transactions

2. Register new user

   https://alexaskill.pythonanywhere.com/?action=register&id=[ALEXA_ID]&btc=foo&eth=bar&l

tc=ayy&doge=ayy2
it will register the [EMAIL_ID] with the btc address foo, eth address bar, ltc address ayy, doge address ayy2 on our databse and then can be used by above API features. Registering again an already registered email will update the fields. Don't use this outisde our website forms @ http://mycryptoportfolio.netlify.com/

How to setup the API?

- run `databse.py` to make the database
- run `python flask_app.py` to start the app
- Upload the databse along with the flask app to pythonanywhere
- configure the app and database both over pythonanywhere server
- setup ports and https endpoint

Requirements :-

- python3
- sqlite
- sqlitebrowser (optional to view the db schema in a GUI)

Pip install :- - flask <<<<<<< HEAD - jsonify - requests - flask_cors

- flask_api

- json
- requests >>>>>>> facbff95e19e8ada55f236a083a73a04b853c9fe