

ALEXA SKILLS

T A B L E O F C O N T E N T

1. Introduction	1
2. Libraries used.....	1
3. Modules	2
3.1. Skill service.....	2
3.2. Skill Interface.....	3
4. Output	4

1. Introduction:

A python program on Alexa Skills for crypto currency transactions.

Here python will accept transactions in CSV format and create a SQLite table in a local database having inserted all the transactions into the table.

User will request to Alexa, that request is converted into a SQL query and fired on the table, then the result is spoken back to user by Alexa.

2. Python Libraries Used:

sqlite3 : Database

csv : to read the csv (transactions)

flask : its a web framework, which will create a localhost server

flask_ask : it is used to send and receive request to-from Alexa

Other program:

Ngrok : securely tunnels the request from Alexa to localhost (Flask).

Amazon Developer console : Alexa Skills set.

3. Alexa skills has two modules:

1. Skill Service
2. Skill interface

3.1 Setup Module 1

1: setup https server

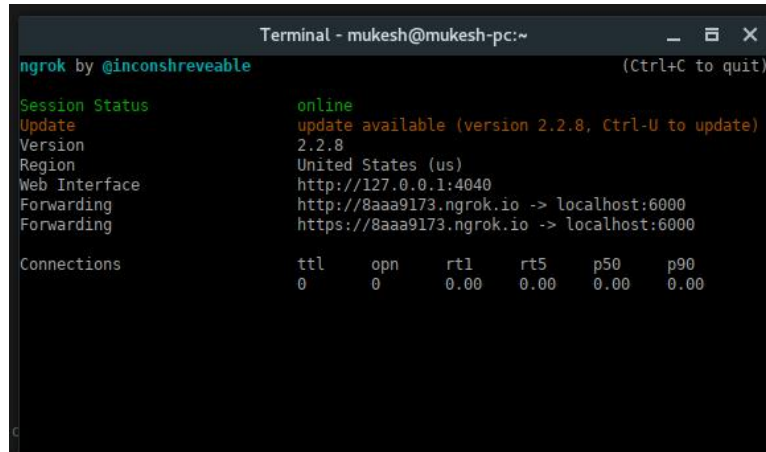
Command: `python cryptAlexa.py`

```
* Running on http://127.0.0.1:6000/ (Press CTRL+C to quit)
* Restarting with stat
* Debugger is active!
* Debugger PIN: 988-472-923
```

Now, Flask localhost server is running on port 6000

2: run ngrok to route the alexa request from internet to this localhost(flask)

Command: `sudo ngrok http 6000`



```
Terminal - mukesh@mukesh-pc:~
ngrok by @inconshreveable (Ctrl+C to quit)

Session Status      online
Update              update available (version 2.2.8, Ctrl-U to update)
Version             2.2.8
Region              United States (us)
Web Interface        http://127.0.0.1:4040
Forwarding            http://8aaa9173.ngrok.io -> localhost:6000
                    https://8aaa9173.ngrok.io -> localhost:6000

Connections
  ttl    opn    rt1    rt5    p50    p90
    0     0     0.00  0.00  0.00  0.00
```

Ngrok will route all request made on <https://8aaa9173.ngrok.io> to localhost:6000
<https://8aaa9173.ngrok.io> will be used in Alexa module 2

3.2. Setup Module 2:

Here we setup Alexa Interface on developer.amazon.com

<https://8aaa9173.ngrok.io> is specified to the Global Endpoints

The screenshot shows the 'Global Fields' configuration page in the Alexa Developer Console. On the left, a sidebar contains a list of configuration sections: Skill Information, Interaction Model, Configuration, SSL Certificate, Test, Publishing Information, and Privacy & Compliance. Each section has a status icon (green checkmark for completed, grey checkmark for pending, and a red 'X' for failed). The 'Configuration' section is highlighted in orange. Below this list is a 'Skills Beta Testing' section with a 'NEW' tag and a status of 'Not yet eligible'. The main content area is titled 'Global Fields' and includes a note: 'These fields apply to all languages supported by the skill.' Under the 'Endpoint' heading, there are two radio buttons for 'Service Endpoint Type': 'AWS Lambda ARN (Amazon Resource Name)' (which is the 'Recommended' option) and 'HTTPS'. The 'HTTPS' option is selected. Below the radio buttons, there is a text input field for the 'Default' endpoint, which contains the URL 'https://cd92f96b.ngrok.io'. At the bottom, there is a question 'Provide geographical region endpoints?' with 'Yes' and 'No' radio buttons; 'No' is selected.

Utterance : if user says the mentioned phrases then Alexa will call the intent named on left-side of the phrase, Which in-turns call the method associated to that intent

Sample Utterances

These are what people say to interact with your skill. Type or paste in all the ways that people can invoke the intents. [Learn more](#)

Up to 3 of these will be used as Example Phrases, which are hints to users.

```
1 AskCoinPrice what is the profit of {coinName}
2 AskCoinPrice the status of {coinName}
3 AskCoinPrice price of {coinName}
4 AskCoinPrice position of {coinName}
5 availQuantity how much {coinName} are available
6 availQuantity quantity of {coinName}
```

4. OUTPUT:

UserID	Date	Coins	BuySell	Qty	Value	Total
mukesh	1-Jan-18	ETC	BUY	100	10	1000
mukesh	1-Jan-18	TRX	BUY	100	10	1000
mukesh	1-Jan-18	TRX	BUY	100	12	1200
mukesh	2-Jan-18	ETC	SELL	-100	11	-1100
mukesh	2-Jan-18	TRX	SELL	-100	11	-1100
mukesh	3-Jan-18	ETC	BUY	100	12	1200
mukesh	3-Jan-18	TRX	SELL	-90	10	-900
mukesh	4-Jan-18	ETC	SELL	-100	15	-1500

The screenshot displays the Alexa Developer Console interface for the 'CryptoAlexa' skill. The top navigation bar includes a 'Return to the developer console' link and a status bar indicating 'Test is enabled for this skill'. The main interface is divided into three sections: 'Alexa Simulator', 'Manual JSON', and 'Voice & Tone'. The 'Alexa Simulator' section on the left shows a chat window with a conversation history. The 'Manual JSON' section in the center displays the JSON input and output for the skill. The 'Voice & Tone' section on the right shows the Echo Show Display, which provides visual feedback for the skill's responses. The chat window shows a series of interactions: the user asks 'open my wallet', the skill responds with 'Hi mukesh, how can I help?'; the user asks 'profit of trx', the skill responds with 'Sorry, you are in loss of 200 for TRX coin'; the user asks 'position of ada', the skill responds with 'You are in profit of -400 for ADA coin'; and the user asks 'quantity of trx', the skill responds with 'You've got 10 TRX coin'. The JSON I/O panel shows the corresponding JSON input and output for each request. The Echo Show Display shows the visual output for the skill, including text and audio responses.

Project Link:

GITHUB: <https://github.com/SutharMukesh/CryptoAlexa/>