NodeJS Index Coding Task *2 hours*

At CryptoCompare, data is our core business and in this exercise we would like you to use data from our services to implement an index calculation engine.

We will be looking for:

- Clear intent of code
- NodeJS code only
- At least **one** unit test for logic in the application (we prefer mochaJs https://www.npmjs.com/package/mocha, but not a requirement)

First, create a CryptoCompare API key using:
https://min-api.cryptocompare.com/documentation and use this key when making connections to the websocket and API.

Task
Websocket documentation page:
https://min-api.cryptocompare.com/documentation/websockets?key=Channels&cat=Trade

1. Subscribe to **BTC-USD** trades from Coinbase, Bitstamp, Kraken, Itbit and Gemini
2. Every time a new trade is received, calculate the penalised volume weighted average price (VWAP) using the last trade from each exchange. Use the last price (“P" field in JSON format message) from each exchange and the volume associated with that price (“Q" field in JSON format message).
3. If the timestamp of the trade is within the last 5 minutes the penalty is 1 (no penalty)
   If the timestamp of the trade is within 5 and 10 minutes old the penalty is 0.8, and so on

See formulas below

$$VWAP = \frac{\sum price * penalty * volume}{\sum penalty * volume}.$$

Time penalty

$$\gamma_t^e := \begin{cases} 1 & \text{if } \tau_t^e < 5 \\ 0.8 & \text{if } 5 \leq \tau_t^e < 10 \\ 0.6 & \text{if } 10 \leq \tau_t^e < 15 \\ 0.4 & \text{if } 15 \leq \tau_t^e < 20 \\ 0.2 & \text{if } 20 \leq \tau_t^e < 25 \\ 0.001 & \text{otherwise} \end{cases}$$

Upload your solution to a private git repository on GitHub, and share it with:
jbutler@cryptocompare.com