Account Login Information and API Keys:

1. OKEK (does not serve US customers)
2. Bitfinex (professional platform requires 10,000 minimum deposit)
3. **GDAX** has been installed and key finalized
4. **Binance** (login: [aprevic@andrew.cmu.edu](mailto:aprevic@andrew.cmu.edu)), MSCF2018, 17762752,
   1. <https://github.com/binance-exchange/binance-official-api-docs>,
   2. <http://python-binance.readthedocs.io/en/latest/binance.html>, <https://github.com/binance-exchange/python-binance>
5. Keyname:
   1. **vniohy1goai**
   2. API Key:

**hsapUzDi0nLhAIcAjp8b56JzQ5YUlpEdr6qquQnY0EvpHgc3ChRdZ9LxA2JEnC2P**

* 1. Secret:

**ITm1NWouloBWlJuzyJKkF7qiJZTKktaA7rnc6wL9SRMYAq4J5jqmrrHXZd2pNuOE**

1. **Kraken**
   1. USERNAME: aprevic
   2. PASSWORD: MSCF2018!
   3. <https://www.kraken.com/help/api>
2. **Bit-Z**
   1. User code: 1210351
   2. [aprevic@andrew.cmu.edu](mailto:aprevic@andrew.cmu.edu) (username?)
   3. MSCF2018 (password)
   4. <https://www.bit-z.com/api.html> (API codes)
3. **Bitstamp**
   1. Client ID: **nopv1225**  
      Password: sMSCF2018 Old one was **fzGV/2mabg?v**
   2. <https://www.bitstamp.net/api/>
   3. Authentication Key

#### NONCE

Nonce is a regular integer number. It must be increased with every request you make. Read more about it [here](http://en.wikipedia.org/wiki/Cryptographic_nonce). Example: if you set nonce to 1 in your first request, you must set it to at least 2 in your second request. You are not required to start with 1. A common practice is to use [unix time](http://en.wikipedia.org/wiki/Unix_time) for that parameter.

#### SIGNATURE

Signature is a HMAC-SHA256 encoded message containing nonce, customer ID (can be found [here](https://www.bitstamp.net/account/balance/)) and API key. The HMAC-SHA256 code must be generated using a secret key that was generated with your API key. This code must be converted to it's hexadecimal representation (64 uppercase characters).

A short code example on how to generate a signature can be seen here:

| **Python** |
| --- |
| import hmac import hashlib  message = nonce + customer\_id + api\_key signature = hmac.new(     API\_SECRET,     msg=message,     digestmod=hashlib.sha256 ).hexdigest().upper() |