Module **AlphaProof**

Classes

```
class Client (api_key, base_url)
```

AlphaProof Client Module

This module allows human and algorithmic traders to create immutables proof of their performance.

The Proof-of-ROI protocol uses a commit-reveal scheme to ensure that the valuable signals are only publicly verifiable once they lost their value.

Dependencies

- requests
- hashlib

Usage

At the moment the signal is created call the `commit` or the `commit_encrypted` method and provide the required data.

After enougth time is passed and the signal has lost its value call the `reveal` or `reveal_encrypted` method.

Examples

Example of how the Module is used can be found at:

`test.py`

Methods

```
def commit(self, signal, price_open, exchange, time_open)
```

Sends the data to be hashed and commited, data is stored on server for later reveal.

Args

signal : str
    the signal either BUY, SELL or STOP

price_open : str
    the price at which the trade was opend

exchange : str
    the exchange on which the trade was executed ("binance" | "bitstamp")

time_open : str
    the time at which the trade was opend, compatible with
    https://pypi.org/project/dateparser/

Returns

a JSON object in format {

    'tx_hash': the transaction identifier from the Ethereum Blockchain,

    'message': 'Success' | error_message

}

```
def commit_encrypted(self, signal, price_open, exchange, time_open)
```

Encrypts and sends the data to be commited. Encrypted message is stored on server for later reveal.

Args

signal : str
the signal either BUY, SELL or STOP

price_open : str
the price at which the trade was opend

exchange : str
the exchange on which the trade was executed ("binance" | "bitstamp")

time_open : str
the time at which the trade was opend, compatible with https://pypi.org/project/dateparser/

Returns

the encryption key used to generated the encrypted hash

a JSON object in format {

'tx_hash': the transaction identifier from the Ethereum Blockchain,

'index': position of where in the commit list it was stored IMPORTANT: this might change when reveals are made, use `open_commits` instead

'message': 'Success' | error_message

}

```
def open_commits(self)
```

Gets a list of all commits that have not yet been revealed

Returns

```
a JSON object in format {
    'list': the list of dicts with commited data,
    'message': 'Success' | error_message

}
```

```
def reveal(self, index=0)
```

Reveals a previous commit by publishing its data, default is index 0 (oldes commit).

Args

`index` : `int` , default= `0`
the index of where the commit is in the list returned by `open_commits`

Returns

```
a JSON object in format {
    'tx_hash': the transaction identifier from the Ethereum Blockchain,
    'message': 'Success' | error_message

}
```

```
def reveal_encrypted(self, encryption_key, signal, price_open, exchange, time_open,
                     index=0)
```

Reveals and encrypted commit by publishing the cleartext data, default is index 0 (oldes commit).

Args

`encryption_key` : `str`
the key returend by `commit_encrypted`

`signal` : `str`
the signal either BUY, SELL or STOP

`price_open` : `str`
the price at which the trade was opend

`exchange` : `str`
the exchange on which the trade was executed ("binance" | "bitstamp")

`time_open` : `str`
the time at which the trade was opend, compatible with https://pypi.org/project/dateparser/

`index` : `int` , default= `0`
the index of where the commit is in the list returned by `open_commits`

Returns

a JSON object in format {

'tx_hash': the transaction identifier from the Ethereum Blockchain,

'message': 'Success' | error_message

}