Wallet Token Stats

You need to write backend for the simple frontend that we have carefully prepared for you.

Run the following command to see how it looks like

```
python3 server.py
```

and open http://localhost:8050 in your browser.

What you need to do is write function `get_data` in `utils/data.py` that extracts statistics for wallet-token pair and plots `Trade history` and `PNL history` graphs.

P.S. Unfortunately gitflic doesn't support latex formulas, so we generated README.pdf from this README.md for you.

Statistics description

`Trade history` is the plot that has `block_number` as x-axis and quote in WETH (a.k.a. `native_quote`) as y-axis and has vertical lines that represent in-transfers (green vertical lines) and out-transfers (red vertical lines).

For you fun you can find wallets with good perfomance (buy low, sell high). We call them sheikhs =). You can come up with your own approach for identification of such wallets.

`PNL history` is the plot that has `block_number` as x-axis and `unrealized PNL` as y-axis.

Unrealized PNL per wallet-token pair

We suggest to use the following formula for `unrealized PNL`

$$\sum_{i=1}^{N} b_{t_i}(p_{t_{i+1}} - p_{t_i}),$$

where b_t is the token balance of the given wallet as of block t, p_t is the `native_quote` quote as of block t, the sequence $\{t_i\}_{i=1}^{N}$ are the blocks at which token transfers occured and t_{N+1} is the latest block for which we calculate `unrealized PNL`.

Getting data from SQL database

We prepared Clickhouse sql database which you may use to derive all the neccessary data. We prepared introductory video for your quick start.

If you want to run sql queries from Python code it is reasonable to use clickhouse-driver library via

```
sql_query = "SELECT * from transactions WHERE block_number = 16000000"
client.execute(sql_query)
```

or

```
sql_query = "SELECT * from transactions WHERE block_number = 16000000"
client.query_dataframe(sql_query)
```

where `client` is initialized in `utils/data.py` in `get_data` function.