

Trend terminals

- Задание
- Требования к коду
- Технические требования
- Требования к графической библиотеке
- Требования по терминалам
- Данные от WebSocket соединения
- Терминалы
 - Price терминал
 - Bid/ask amount терминал
 - SL терминал
 - SLA терминал
 - Delta терминал
 - DeltaA терминал
 - SLA_SM таблица
- Подход к реализации
- Результат

Задание

Нужно разработать терминалы, которые будут подключаться к WebSocket серверу, в реальном времени получать данные, производить расчёты и выводить результат в терминал (на график). Должны быть реализованы следующие терминалы с таблицами (в скобках приведены оси терминала - x,y):

- Price терминал (price, time)
- Bid/ask amount терминал (amount, time) - 2 графика в одном терминале
- SL терминал (sl коэффициент, time)
- SLA терминал (sla коэффициент, time)
- Delta терминал (delta значения, time)
- DeltaA терминал (deltaA значения, time)
- SLA_SM таблица

Требования к коду

- Терминалы должны быть разработаны на JavaScript
- Библиотека графиков
 - Выбор библиотеки обсуждается вместе, требования к библиотеке приведены ниже
- Простота и чистота кода
- Комментарии к коду
- Комментарии должны основываться на синтаксисе jsDoc

Технические требования

- Если используются промисы, то использовать синтаксис `async/await`
- Если используется `timestamp`, то он должен быть в миллисекундах

Требования к графической библиотеке

- Библиотека должна быть зрелая, популярная и хорошо задокументирована
- Библиотека должна поддерживать
 - Отображение данных в реальном времени
 - Отображение нескольких графиков одновременно
- Пример библиотеки: TradingView (<https://www.tradingview.com/HTML5-stock-forex-bitcoin-charting-library/>)

Требования по терминалам

1. Терминалы должны отображаться на HTML странице. Один терминал - одна HTML страница. Это нужно для того, чтобы была возможность открыть несколько HTML страниц с терминалами на экране.
2. Терминалы обновляются в реальном времени. Данные приходят по WebSocket соединению.
3. Каждый терминал отображает данные по определенной бирже и определенной криптопаре (символу). В терминале должна быть возможность выбора биржи и символа. Названия бирж и символов можно получить в данных `snapshot'a`, см. "Данные от WebSocket соединения".
4. Ось X является всегда временем. Время в секундах.
5. Терминалы отображают данные за последние n-секунд.
6. Возле терминала должна располагаться таблица с последними 10-100 значениями, количество значений зависит от того, насколько будет удобно пользоваться этой таблицей. К примеру возьмём `price` терминал: если значений 100, это означает, что в таблице отображаются данные за последние 100 секунд, первая колонка - секунды, вторая - цены. Таблица также должна

обновляться в реальном времени как и график.

7. К таблице с данными должен быть реализован ввод данных (2 значения: от и до). К примеру: к таблице с ценами можно ввести промежуток цены (от и до). Если значения будут удовлетворять этому промежутку, тогда эти данные должны подсвечиваться зеленым цветом. Если значение "от" задано, а значение "до" нет, это означает, что нужно подсвечивать все записи от данного значения. Если значение "до" задано, а значение "от" нет, это означает, что нужно подсвечивать все записи до данного значения.

Данные от WebSocket соединения

Подключиться к WebSocket серверу можно по следующей ссылке: `ws://178.62.232.211:6999`

Если подключение не происходит, сообщите нам, возможно на сервере ведутся работы или сервер просто отключён.

Описание данных, которые будут приходить от сервера, найдёте в приложенном pdf файле.

Пример кода подключения к серверу (Node.js реализация):

```

'use strict';

const WebSocket = require('ws');

(() => {

  const commands = {
    subscribe: {
      command: 'subscribe',
      payload: {
        seconds: 10 // get metric data of the last 10 seconds
      }
    },
    unsubscribe: {
      command: 'unsubscribe'
    }
  };

  const ws = new WebSocket('ws://178.62.232.211:6999');

  ws.on('open', () => {
    ws.send(JSON.stringify(commands.subscribe));
  });

  ws.on('message', message => {
    console.log(JSON.stringify(JSON.parse(message), null, 4));
  });

  ws.on('error', error => {
    console.log('Error occurs in WebSocket connection:');
    console.log(error);
  });

  ws.on('close', () => {
    console.log('Connection is closed');
  });

})();

```

На данный момент WebSocket сервер посылает данные по 3-м биржам: Bitfinex, Binance и HitBTC. От каждой биржи посылаются данные по следующим криптопарам: BTCUSD, ETHUSD.

Терминалы

В данном разделе приведено более полное описание терминалов и формулы расчётов. Примеры данных терминалов приведены в xls файле.

Price терминал

Price терминал отображает график зависимости цены от времени. Необходимые данные приходят по WebSocket соединению.

График отображает price значения за каждую секунду.

Bid/ask amount терминал

Bid/ask amount терминал отображает график зависимости bid amount и ask amount от времени. Необходимые данные получаем по WebSocket соединению. В терминале должно отображаться 2 графика: bid amount от времени и ask amount от времени.

Графики отображают bid amount и ask amount за каждую секунду.

SL терминал

SL терминал отображает график зависимости sl коэффициента от времени. SL коэффициент высчитывается по формуле:

$$SL = bid\ amount / ask\ amount$$

Bid amount и ask amount получаем по WebSocket соединению.

График отображает коэффициенты SL за каждую секунду.

SLA терминал

SLA терминал отображает график зависимости sla коэффициента от времени. Этот коэффициент означает среднее значение SL коэффициентов за определенный промежуток времени (за определенный промежуток секунд). Количество секунд должно задаваться вручную в терминале. SLA коэффициент высчитывается по формуле:

$$SLA = \text{SUM}(SL_1, SL_2, SL_3, \dots, SL_N) / N, \text{ где } N - \text{количество секунд}$$

Коэффициенты SL_1, SL_2, SL_3, ..., SL_N высчитываются по формуле указанной в разделе "SL терминал"

График отображает коэффициенты SLA за каждые N секунд.

Delta терминал

Delta терминал отображает график зависимости delta значений от времени. Delta значение - это разница между bid amount и ask amount или наоборот. Формула:

$$Delta = bid\ amount - ask\ amount$$

или

$$Delta = ask\ amount - bid\ amount$$

В терминале должна быть возможность выбора подсчёта delta (подсчёт по первой формуле или по второй формуле).

График отображает значения delta за каждую секунду.

DeltaA терминал

DeltaA терминал отображает график зависимости deltaA значений от времени. DeltaA высчитывается по формуле:

$$DeltaA = \text{SUM}(bid\ amount\ 1, \dots, bid\ amount\ N) - \text{SUM}(ask\ amount\ 1, \dots, ask\ amount\ N)$$

или

$$DeltaA = \text{SUM}(ask\ amount\ 1, \dots, ask\ amount\ N) - \text{SUM}(bid\ amount\ 1, \dots, bid\ amount\ N)$$

где N - количество секунд, которое должно задаваться вручную в терминале.

Другими словами, DeltaA - это разница между суммой bid объёмов и суммой ask объёмов за определенный промежуток времени.

В терминале должна быть возможность выбора подсчёта deltaA (подсчёт по первой формуле или по второй формуле).

График отображает значения deltaA за каждые N секунд.

SLA_SM таблица

SLA_SM таблица - это таблица с 2 колонками. Первый столбец отображает номер секунды, второй - значение SLA_SM в процентах. Высчитывается по следующей формуле:

$$SLA_SM = (SLA_1 / (SUM(SLA_2, SLA_3, SLA_4, SLA_5, SLA_6) / 5) - 1) * 100$$

Другими словами, это отношение SLA за последнюю секунду к среднему значению SLA за 5 предпоследних секунд.

Подход к реализации

1. Имплементировать подключение к WebSocket соединению и получение данных.
2. Реализовать 1 терминал с таблицей.
3. Протестировать реализованный терминал, отрефакторить, откомментировать.
4. Реализовать похожим способом остальные терминалы.

Результат

Результат работы предоставить в git репозитории с комментариями и инструкциями запуска, инструкции могут быть описаны прямо в README.md файле репозитория.