Trade monitor

Нужно разработать приложение, которое будет подписываться на получение трейдов (выполненных) по WebSocket соединению, сохранять их в базу данных и предоставлять простой графический интерфейс для скачивания файла в xls формате.

Данные трейдов, которые нам необходимы:

- timestamp - время, когда трейд был выполнен
- price - цена, по которой выполнился трейд
- amount - объём трейда
- type - тип трейда (buy или sell)

Таблица в базе данных, которая будет содержать эти данные, будет иметь следующие поля:

- id - primary key
- exchange_name - имя биржи, строковый тип
- symbol - название символа, строковый тип
- timestamp - timestamp (дата, когда ордер исполнился)
- price - вещественный тип
- amount - вещественный тип
- type - будет хранить значение 1 (buy) или значение 2 (sell)

Ход работы

При запуске приложения, приложение открывает WebSocket соединения с биржами, подписывается на трейды определённых символов (смотреть конфиг ниже). Биржи начнут посылать данные, эти данные нужно сохранять в базу данных. В бирже нужно хранить данные за последние N секунд (N секунд берётся из конфига - см. ниже).

При прерывании соединения, должно быть реализовано автоматическое переподключение. Если данные с биржи не приходят в течении определенного количества секунд (см. ниже в конфиге), производить переподключение.

Методы

Нужно разработать метод получения выполненных трейдов через WebSocket соединение следующих бирж:

- Binance (исходники этой биржи у вас имеются, но мы вам их всё равно предоставим, т.к. там были корректировки)
- Bitmex (исходники данной биржи мы предоставим)

Семантика метода:

```
/**
  * Subscribe to trades via WebSocket connection
  * @param {Array<string>} symbols - array of exchange symbols
  * @return {Promise<Array<string>|Error>}
  */
async subscribeTrades(symbols) {}
```

Данный метод должен возвращать промис с массивом подписанных символов. При неудачной подписке, возвращает промис с объектом Error.

Интерфейс

Нужно разработать простой интерфейс, в котором будут следующие поля:

- Поле выбора биржи - dropdown box
- Поле выбора символа - dropdown box
- Поле ввода секунд - числовое input поле
- Кнопка “Скачать”

После выбора биржи, символа и ввода секунд, пользователь нажимает на кнопку “Скачать“ и происходит скачивание файла в xls формате. В файле будут находится данные (трейды) за последние N секунд (сколько было введено в input поле).

Файл будет содержать следующие поля:

- Exchange name - название биржи
- Symbol - название символа

- Date - дата (формат задаётся в конфиге ниже)
- Price - цена
- Amount - объём
- Order type - тип ордера (“buy” или “sell“)

Ошибки

При возникновении ошибок, логировать их через логгер Winston (https://www.npmjs.com/package/winston)

Путь для логирования: ./logs/tradeMonitor.log

Конфиг

Название файла: config.js

Будет содержать следующие значения:

```
module.exports = {
    mode: 'dev',
    port: 8080,
    seconds: '3600',
    utcOffset: 120,
    timeFormat: 'YYYY-MM-DD HH:mm:ss.SSS',
    inactivityTimeout: {
      Binance: 3000,
      Bitmex: 3000,
    },
    trades: {
      Binance: ['BTCUSDT'],
      Bitmex: ['XBTUSD'],
    },
}
```

Описание:

- port - порт на котором будет висеть приложение
- seconds - за какое количество времени хранить данные
- utcOffset - сдвиг от UTC времени (в минутах)
- timeFormat - формат времени
- inactivityTimeout - если в течении данного количества милисекунд не приходят данные с биржи, производить переподключение по WebSocket соединению
- trades - символы на которые нужно подписаться, также эти данные можно использовать для вывода в интерфейсе в dropdown полях

Технологии

- Node.js v12.13.1
- MySQL (https://www.npmjs.com/package/mysql2) - через async/await
- Использовать async/await
- Для работы с датой использовать moment (https://www.npmjs.com/package/moment)
- Winston логгер (https://www.npmjs.com/package/winston)
- Express framework (https://www.npmjs.com/package/express)

Результат

Результат работы предоставить в git репозитории с инструкциями по установке.