Санкт-Петербургский государственный университет

Кафедра системного программирования

Решетников Илья Алексеевич

Разработка модуля для подключения к потоковому API валютной биржи

Отчет по учебной практике

Научный руководитель:
доцент кафедры СП, к. т. н. Литвинов Ю. В.

Консультант:
глава отдела разработки ФЛ«ДИЭСЭКС ТЕХНОЛОДЖИЗ РАША»
Зубаревич Д.А.

Санкт-Петербург
2021

Оглавление

Введение

Компания DSX Technologies занималась разработкой финансового программного обеспечения. Основным проектом компании являлась валютная биржа "digital securities exchange".

На бирже DSX поддерживались операции над криптовалютами: Bitcoin, Litecoin, Ripple, EOS, Ether, Bitcoin Cash, Bitcoin Gold, EURS, USDT, Bitcoin SV, фиатными валютами: RUB, USD, EUR, GBP.

Для поддержания конкурентоспособности бирже необходимо предоставлять своим клиентам достойный уровень сервиса. Частью этого сервиса является модуль для подключения к потоковому API биржи DSX, разработка которого является целью данной работы. Данный модуль позволил бы клиентам создавать программное обеспечение для автоматизации получения данных с биржи. А для DSX наличие такого модуля могло способствовать привлечению новых клиентов.

К сожалению, на данный момент биржа не осуществляет финансовые операции. 04 декабря 2020 года началась процедура банкротства биржи [5]. Причиной, которая привела DSX к банкротству, является приостановка работы платежной системы ePayments [4].

Постановка задачи

В рамках данной работы были поставлены следующие задачи.

- Разработка модуля для подключения к потоковому API биржи.
- Тестирование разработанного модуля.

Обзор используемых технологий

Разработка модуля для подключения к потоковому API биржи представляла собой расширение существующей открытой библиотеки XChange-stream [2], по этой причине в разработке модуля использовались технологии, которые лежат в основе библиотеки.

XChange-stream обеспечивает простой и согласованный потоковый API для взаимодействия с биржами через протокол WebSocket [7], предназначенный для обмена сообщениями в режиме реального времени.

Библиотека XChange-stream позволяет пользователям подписываться на обновления в реальном времени с помощью реактивных потоков библиотеки RxJava [1], предназначенной для создания асинхронных программ и программ, основанных на событиях, с использованием наблюдаемых последовательностей.

Данная библиотека распространяется по лицензии Apache [6] – это позволяет пользователям свободно встраивать ее в свои приложения, а бирже выгодным образом увеличить потенциальное количество пользователей.

Реализация

В рамках взаимодействия с биржей DSX были добавлены классы для создания подключения к бирже по протоколу WebSocket, получения и обработки данных таблицы лимитных заявок, осуществленных сделок и цены валюты в момент времени для заданной валютной пары.

Диаграмма зависимостей ключевых классов модуля отображена на (рис. 1)

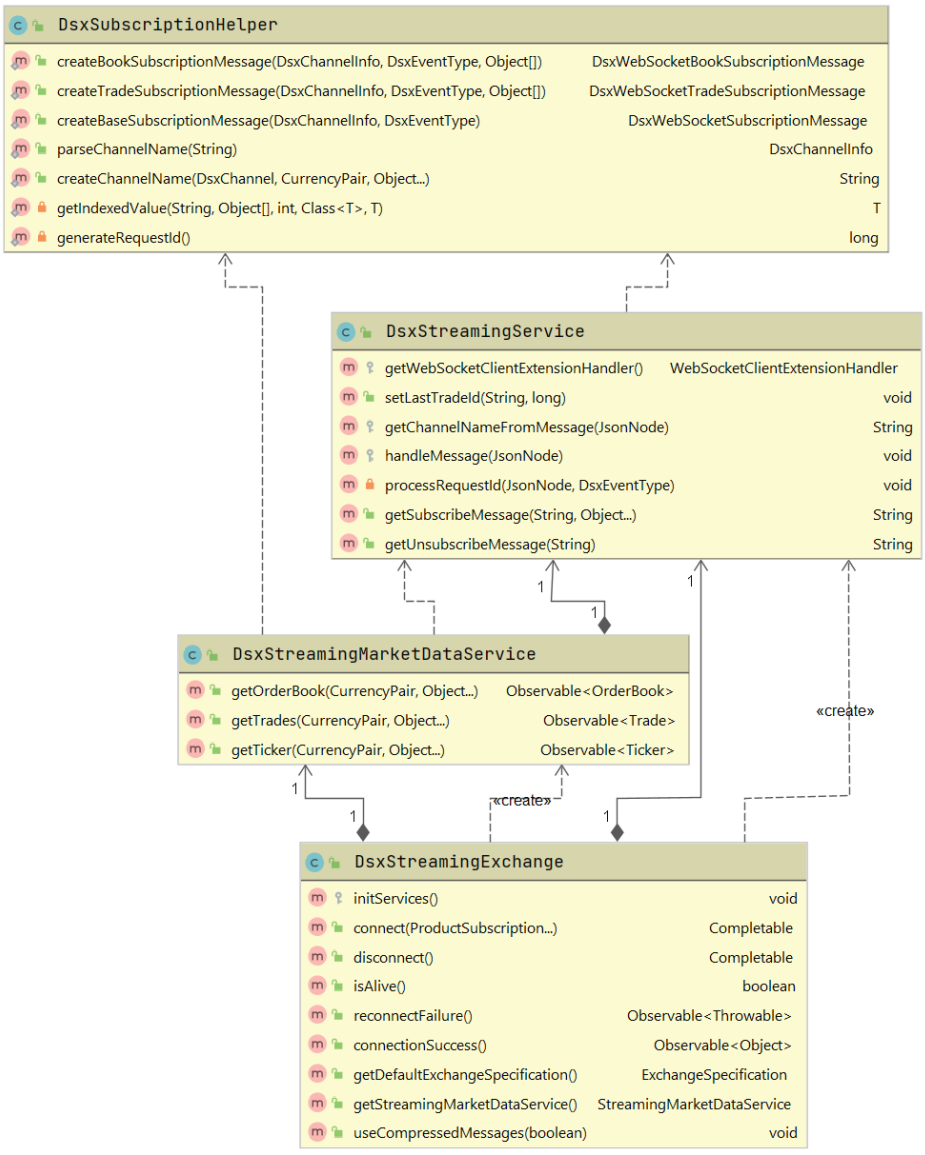

Рис. 1: Диаграмма классов модуля

Описание ключевых классов модуля

- DsxStreamingExchange – представляет собой оболочку над классами DsxStreamingMarketDataService, DsxStreamingService.
- DsxStreamingMarketDataService – реализует методы получения и обработки потоков данных. В качестве примера на листинге 1 приведен метод получения и обработки потока данных о сделках, заключенных на бирже.

```
65  public Observable<Trade> getTrades(CurrencyPair currencyPair, Object... args) {
66    String channelName = createChannelName(DsxChannel.trade, currencyPair, args);
67    final ObjectMapper mapper = StreamingObjectMapperHelper.getObjectMapper();
68    Observable<JsonNode> jsonNodeObservable = service.subscribeChannel(channelName, args);
69    return jsonNodeObservable
70      .map(jsonNode -> mapper.readValue(jsonNode.toString(), DsxTradeMessage.class))
71      .map(DsxTradeMessage::getTrades)
72      .flatMap(trades ->
73        Observable.fromIterable(
74          Arrays.stream(trades)
75            .sorted(Comparator.comparingLong(DSXTrade::getTid))
76            .collect(Collectors.toList())
77        )
78      )
79      .doOnNext(trade -> service.setLastTradeId(channelName, trade.getTid()))
80      .map(trade -> DSXAdapters.adaptTrade(trade, currencyPair));
81    }
```

Листинг 1: Метод получения и обработки сделок

Исходя из листинга видно, что в первую очередь происходит генерация названия канала и подписка на него. В качестве обработки данных происходит преобразование полученных данных к общему для библиотеки классу Trade, а в классе DsxStreamingService обновляется значение идентификатора последней сделки. Аналогичным способом определяются методы для таблицы лимитных заявок и цены валюты в момент времени.

- DsxStreamingService – представляет собой WebSocket-клиент. Отвечает за взаимодействие с сервером: поддержку соединения, получение и отправку сообщений.
- DsxSubscriptionHelper – представляет собой вспомогательный класс, который отвечает за генерацию сообщений для подписки на потоки данных.

Для проверки корректности работы модуля были созданы два класса.

- DsxStreamingMarketDataServiceTest – класс, содержащий модульные тесты для проверки корректности методов получения таблицы лимитных заявок и совершенных сделок. Корректность проверялась путем проверки равенства значений до преобразования и после.

- DsxManualExample – представляет собой пример программы, которая отражает корректное использование модуля: создание соединения, получение данных по каналам. Данный пример указан на листинге 2.

```
public class DsxManualExample {
  private static final Logger LOG = LoggerFactory.getLogger(DsxManualExample.class);

  public static void main(String[] args) throws InterruptedException {
    ExchangeSpecification specification = new ExchangeSpecification(DsxStreamingExchange.class);
    specification.setExchangeSpecificParametersItem(
    DsxStreamingExchange.DSX_SPEC_PARAMS_API_URI,
    "ws://localhost:8080/stream");
    specification.setShouldLoadRemoteMetaData(false);
    StreamingExchange exchange = StreamingExchangeFactory.INSTANCE.createExchange(specification);

    exchange.connect().blockingAwait();

    Disposable orderBookObserver = exchange.getStreamingMarketDataService()
    .getOrderBook(CurrencyPair.BTC_USD, DsxInstrumentType.LIVE, 1)
    .subscribe(orderBook -> {
        LOG.info("First ask: {}", orderBook.getAsks().get(0));
        LOG.info("First bid: {}", orderBook.getBids().get(0));
    }, throwable -> LOG.error("Error in getting order book: ", throwable));

    Disposable tradeObserver = exchange.getStreamingMarketDataService()
    .getTrades(CurrencyPair.BTC_USD, DsxInstrumentType.LIVE)
    .subscribe(trade -> {
        LOG.info("Trade: {}", trade);
    }, throwable -> LOG.error("Error in getting trade: ", throwable));

    Disposable tickerObserver = exchange.getStreamingMarketDataService()
    .getTicker(CurrencyPair.BTC_USD, DsxInstrumentType.LIVE)
    .subscribe(ticker -> {
        LOG.info("Ticker: {}", ticker);
    }, throwable -> LOG.error("Error in getting ticker: ", throwable));

    Thread.sleep(10000);
    orderBookObserver.dispose();
    tradeObserver.dispose();
    tickerObserver.dispose();

    exchange.disconnect().subscribe(() -> LOG.info("Disconnected"));
  }
}
```

Листинг 2: Пример использования модуля

Успешное создание соединения и получение данных в классе DsxManualExample, а так же верные преобразования данных, протестированные в классе DsxStreamingMarketDataServiceTest позволяют сделать вывод о том, что разработанный модуль корректно работает.

Заключение

В ходе данной работы был разработан и протестирован модуль для подключения к потоковому API биржи DSX. На момент закрытия биржи модуль был готов к введению к в эксплуатацию, но в силу обстоятельств этого не произошло.

Код данной работы представлен по ссылке [3].

Список литературы

[1] GitHub. RxJava. — 2021. — URL: https://github.com/ReactiveX/RxJava.

[2] GitHub. XChange-stream. — 2021. — URL: https://github.com/bitrich-info/xchange-stream.

[3] GitHub. XChange-stream-dsx. — 2021. — URL: https://github.com/dsx-tech/xchange-stream/tree/dsx-stream/xchange-dsx.

[4] Magnates Finance. FCA Orders Epayments Systems to Suspend Operations on AML Concerns. — 2020. — URL: https://www.financemagnates.com/fintech/payments/epayments-systems-suspends-operations-on-aml-concerns/.

[5] Profinvestment. Биржа криптовалют DSX (DSXglobal). — 2021. — URL: https://profinvestment.com/dsx/.

[6] Wikipedia. Apache License. — 2021. — URL: https://en.wikipedia.org/wiki/Apache_License.

[7] Wikipedia. WebSocket. — 2021. — URL: https://en.wikipedia.org/wiki/WebSocket.