Trend bar service

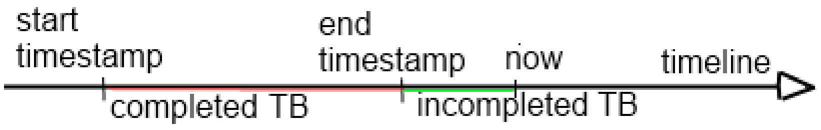
Terminology

Symbol - currency pair (e.g. EURUSD, EURJPY)

Quote - a price update for a particular symbol which happened at certain moment of time. Contains new price, symbol and timestamp.

Trend bar (TB) - also known as candlestick - aggregated quote data for a particular symbol within a given trend bar period. More details about trend bars (candlesticks) can be found here: OHLC chart or candlestick chart. Note that it doesn't contain all information about quotes it's based on, it contains following parameters only:
open price - a price for a given symbol at the beginning of a trendbar period
close price - a price of from the last received quote in this trendbar period
high price - max value of the price during a period
low price - min value of the price during a period
trendbar period - certain time interval during which quotes are accumulated (e.g. M1 - one minute, H1 - one hour, D1 - one day and so on).
timestamp - a moment of time at which trendbar period is started
Trendbar always starts at the beginning of time period. If it's minutely trend bars they are starting at 00:00:00, 00:01:00 and so on. If it's hourly trend bars there are starting at 00:00:00, 01:00:00 and so on.

Completed TB - a TB, time interval of which is over. If, for example, we're considering TB of type M1, then completed TBs are the TBs which has been created before or at the beginning of a previous minute.



TB history - a set of a completed TBs start time of which is in a specific period of time.

General requirements

- Sources must be compilable. If they are not then task is not considered as completed.
- Source code must be covered with tests. JUnit or TestNG could be used as a test framework.
- It's very important to make source code as clean (in terms of Clean code book) and easy to understand as possible. Don't waste your time on Javadoc, good code is supposed to document itself.

- Code should be formatted according to standard Java Code Style(no single line if-s/for-s)

- Project structure should be Maven-compliant and have `pom.xml` file, which describes a project and its dependencies, in a root directory.
- Requirements that are described in **Bonus** section are not obligatory, it's something that's nice to have. They are listed in order of importance and value, from higher to lower.

Task description

Let's assume that there are three types of trendbars. Namely - M1 (minutely), H1 (hourly) and D1 (daily).
It's necessary to implement a trend bar service, which:
1. Builds TBs based upon received quotes. It means that service should maintain a set of current trend bars: update them on each quote and persist them to some storage as soon as a trend bar period of a bar is over.
2. Provides trend bars history by a request. That is it returns a set of trend bars by a given symbol name, trend bar period and *from* timestamp and *to* timestamp. If *to* timestamp is omitted method should return all trend bars that are between *from* timestamp and now.

Limitations and clarifications

TB history includes completed TBs only.
TB storage could be in-memory.
Don't use 3rd party frameworks with exception of Dependency Injection frameworks.
Quotes are coming in a natural order, that is timestamp of a next quote is always bigger than timestamp of a previous one.
Keep in mind that trend bar service is a module of a system, not a standalone application, so there is no need to expose this service via web services or implement a UI interface for it.
The best thing to do is to apply Test Driven Development while you implement this task.
Dependency Injection is very useful approach. Don't hesitate to use it where it's applicable.

Bonuses

TB service should process received quotes asynchronously that is not process a received quote in the same thread, but put it into some kind of internal storage for further processing by another thread.
It would be nice to have trivial implementations of quotes provider to use them in integration tests.
Clients of TB service should be able not only to register in TB service, but unregister as well.