

Тестовое задание для вакансии бэкенд-разработчика: котировки

Общие сведения

Необходимо реализовать серверную часть сервиса для предоставления торговому терминалу текущих котировок и исторических данных для построения графиков.

Основные функции сервиса:

1. Получать текущие котировки по инструментам с сервера котировок;
2. Из полученных котировок обновлять и хранить историю для построения графиков «японские свечи» для всех периодов;
3. Реализовывать возможность получения истории для построения графика frontend-частью;
4. Реализовывать возможность получения текущих котировок для построения таблицы и обновления последней точки на графике.

Создание самого торгового терминала выходит за рамки данного задания.

Описание источника данных приведено ниже.

Предполагается, что фронтенд-часть будет реализована в браузере, используя только HTML5, CSS3, ECMAScript5. Текущие котировки должны получаться с помощью JavaScript, использование Flash исключается. Отрисовка графиков должна происходить с помощью SVG и/или Canvas.

Глоссарий

Котировка — быстро меняющаяся биржевая цена (курс) товара, которую объявляет продавец или покупатель и по которой они готовы совершить покупку или продажу.

Торговый инструмент — товар, по которому ведутся торги на бирже.

Торговый терминал — программа, предназначенная для совершения сделок клиентом на финансовом рынке.

Технические ограничения

Основной инструментарий для разработки — PHP5.

Для отдельных функций можно использовать и другие средства, но обоснованность их использования необходимо будет аргументировать.

Среда выполнения — Linux (поддерживаемый стабильный дистрибутив, например, Ubuntu 16.04 / 14.04 или Debian Jessie).

Сервис должен быть работоспособен на одном узле в течение длительного времени без падений и утечек памяти, с легкостью выдерживая 1000 одновременных запросов клиентов.

Возможность горизонтального масштабирования — плюс, но не обязательна.

Требования

Сервис должен возвращать описанные ниже данные.

1. Котировки инструментов

В терминал должны доставляться список торгуемых финансовых инструментов и котировка (текущая цена) по каждому из них.



Символ	Бид	Аск
EURUSD	1.35883	1.35897
GBPUSD	1.60653	1.60680
USDCHF	0.90312	0.90340
USDJPY	97.123	97.137
AUDUSD	0.94473	0.94500
NZDUSD	0.83303	0.83336
USDCAD	1.03130	1.03157
EURJPY	131.978	132.005
GBPJPY	1.70002	1.70074

Котировка (текущая цена) финансового инструмента в торговое время изменяется до нескольких раз в секунду. Все изменения котировки необходимо доставлять в клиентские торговые терминалы с максимально возможной скоростью.

Одновременно по каждому торговому инструменту существует два значения котировки:

- *Bid* - цена спроса, т.е. цена, по которой брокер предлагает покупку у клиента объема финансового инструмента;
- *Ask* - цена предложения, т.е. цена, по которой брокер предлагает продажу объема финансового инструмента.

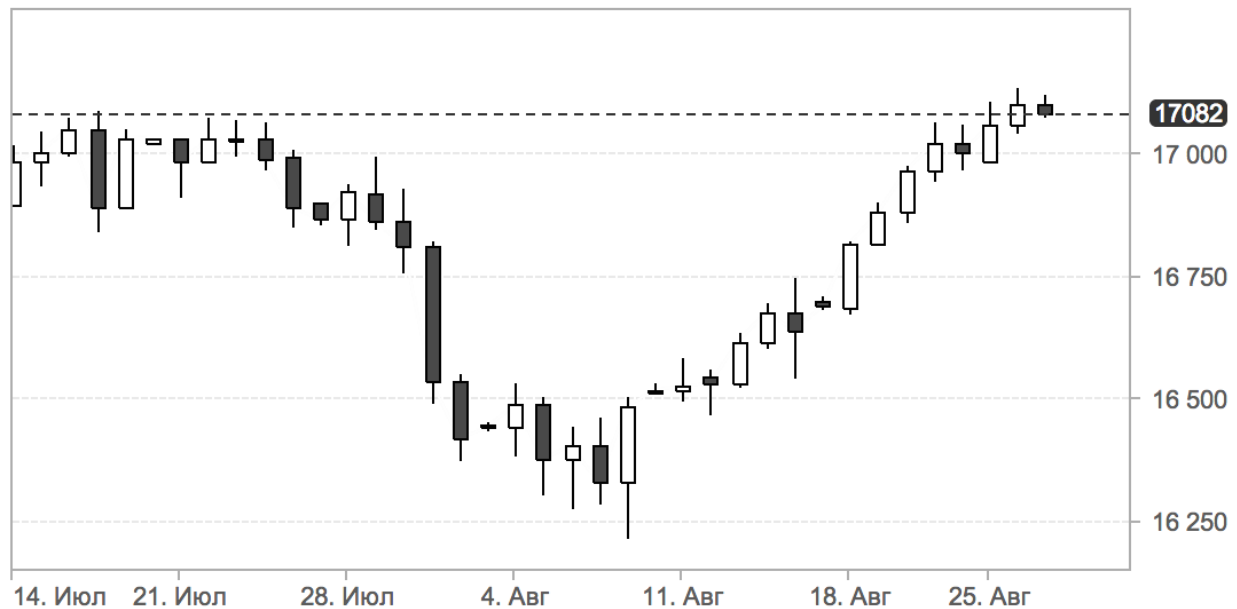
Примечание: для упрощения в задании будет использоваться только значение Bid.

Приоритет — максимально быстрая доставка данных.

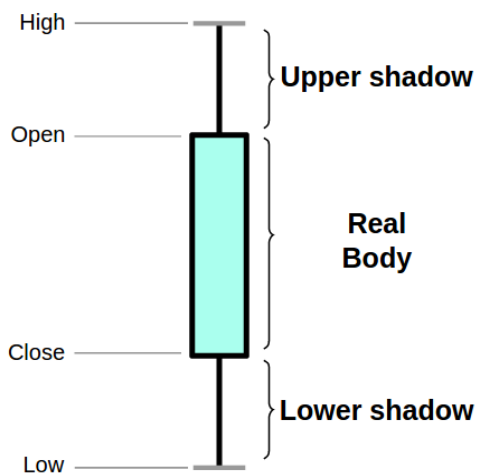
2. Данные для графика цены выбранного инструмента

В сервисе должна быть предусмотрена возможность получения исторических данных для построения графика по выбранному инструменту.

Ниже приведен пример графика «Японские свечи».



2.1. Характеристики свечи



Свеча отражает изменения цены за некоторый фиксированный период времени через параметры:

- *Open* — котировка в момент начала периода;
- *Close* — котировка в момент окончания периода;
- *High* — максимальная котировка за период;
- *Low* — минимальная котировка за период;
- *Time* — время начала периода свечи.

2.2. Периоды построения

Период времени является фиксированным. В торговом терминале используются следующие периоды свечей:

Наименование периода	Продолжительность	Глубина истории
M1	1 минута	1 торговый день
M5	5 минут	3 торговых дня
H1	1 час	10 торговых дней
D1	1 день	∞
W1	1 неделя	∞

Пример для периода M1: каждая свеча отражает движение цены выбранного инструмента в течение одной минуты. Глубина истории — 1 торговый день, то есть нужно вернуть данные по всем свечам M1 за сутки.

2.3. Агрегация и генерация исторических данных

Сервис должен генерировать исторические данные для графиков путем агрегирования моментальных котировок (т. н. «тиков») по заданным периодам.

Для упрощения в данном задании необходимо генерировать исторические данные только по интервалам M1, H1 и D1.

Способ генерации исторических данных следующий:

при получении котировки (моментальной цены) из источника фиксируется ее время и цена.

Для каждого из интервалов (минута, час, день) должны вычисляться и сохраняться следующие значения цен:

- Open — Цена открытия (первая цена за период);
- Close — Цена закрытия (последняя цена за период);
- Low — Минимальная цена за период;
- High — Максимальная цена за период.

Данные параметры для свечей за текущий период (например, за незавершенный час или день) могут постоянно меняться. Для свечей за завершенные периоды данные должны оставаться неизменными.

Часовой пояс — MSK (GMT+4).

Источник данных

Для получения данных необходимо подключиться по TCP/IP к источнику котировок:

IP: 178.62.145.164

порт: 10000

После подключения начнут выдаваться котировки: символ инструмента, цена Bid и время в текстовом виде.

Пример получения данных:

```
telnet 198.211.118.180 10000
```

Результат:

```
S=GBPUSD;T=2016-06-01T13:23:24;B=1.44187
```

```
S=EURUSD;T=2016-06-01T13:23:24;B=1.11662
```

```
S=GBPUSD;T=2016-06-01T13:23:25;B=1.44185
```

```
S=XAUUSD;T=2016-06-01T13:23:26;B=1215.22
```

```
S=XAUUSD;T=2016-06-01T13:23:27;B=1215.12
```

```
S=GBPUSD;T=2016-06-01T13:23:27;B=1.44184
```

...

Каждому инструменту соответствует отдельная строка. Параметры разделены между собой точкой с запятой.

Пояснения:

- **S**=EURUSD — обозначение (символ) инструмента;
- **T**=2016-06-01T13:23:24 — время в формате ISO 8601 (yyyy-mm-ddThh:ii:ss);
- **B**=1.11662 — цена Bid.

Предоставление результатов

В результате выполнения задания необходимо предоставить исходный код (желательно в виде ссылки на [github-](#) или [bitbucket-репозиторий](#)), а также *одно* из нижеуказанного:

1. [Dockerfile](#) для сборки Docker-образа (можно также использовать [Docker Compose](#)) с настроенным окружением приложения (**предпочтительный вариант**);
2. Подробная инструкция по развертыванию сервиса с указанием требований к среде выполнения (тип и версия дистрибутива ОС, список необходимых библиотек с указанием версий, нестандартные настройки и расширения). Приветствуется простота настройки.

Дополнительно может быть предоставлен адрес с уже развернутой копией сервиса.