Тестовое задание Q-SYSTEMS (Д. Вишневский)

Оглавление

[Исходное техническое задание 1](#__RefHeading__329_2132297899)

[Общие комментарии по техническому заданию 2](#__RefHeading__213_2132297899)

[Источник данных о курсах валют 3](#__RefHeading__215_2132297899)

[Структура базы данных 4](#__RefHeading__217_2132297899)

[Структура проекта 5](#__RefHeading__219_2132297899)

[Unit тесты 6](#__RefHeading__331_2132297899)

[Пользовательский интерфейс 7](#__RefHeading__223_2132297899)

[Средства разработки и исходники 8](#__RefHeading__684_1651664074)

* + 1. Исходное техническое задание

Разработать веб-приложение на spring framework.

Веб-приложение должно содержать RESTful интерфейс, пользовательского веб-интерфейс. Пользовательский веб-интерфейс должен состоять из одной страницы – страница, показывающая курсы валют, взятые с БД. Страница с курсами валют — выводит валюты и курсы выбранной валюты на текущую дату и несколько предыдущих.

Для хранения данных можно использовать любую базу данных (например, msSQL). Необходимо ее заполнить валютами и значениями за несколько дней.

То есть пользователь заходит на веб-интерфейс, кликает на желаемую валюту и ему подгружаются данные из БД на текущую дату и несколько предыдущих.

Желательно написать несколько юнит тестов.

Рекомендуемые фреймворки: spring mvc, spring boot, spring data.

Много деталей в ТЗ нет, потому что задание творческое, любые детали можете додумывать сами. С реальными заказчиками бывает так же — они не всегда точно знают, что хотят «сделайте демку — а мы посмотрим».

Что хотелось бы увидеть:

* чистый код, покрытый юнит тестами
* архитектуру, разделенную на слои
* хороший стиль ОО-проектирования, принципы SOLID.

На веб-интерфейсе красота не требуется, можно просто в таблицу выводить всю информацию.

* + 1. Общие комментарии по техническому заданию

**Веб-приложение должно содержать RESTful интерфейс, пользовательского веб- интерфейс. Пользовательский веб-интерфейс должен состоять из одной страницы – страница, показывающая курсы валют, взятые с БД. Страница с курсами валют — выводит валюты и курсы выбранной валюты на текущую дату и несколько предыдущих.**

Я бы сформулировал по другому:

* Страница по умолчанию должна отображать курсы валют доступные на текущую дату
* Должна имеется возможность посмотреть доступные курсы валют на любую выбранную дату
* Для выбранной валюты должна иметься возможность посмотреть историю изменения курса за заданный промежуток времени

**Для хранения данных можно использовать любую базу данных (например, msSQL). Необходимо ее заполнить валютами и значениями за несколько дней.**

Я буду использовать МуSQL, т. к. она у меня установлена. Кроме того spring data позволяет практически полностью абстрагироваться от типа базы данных.

**Рекомендуемые фреймворки: spring mvc, spring boot, spring data.**

Для данного задания не вижу смысла использовать spring mvc при наличии REST интерфейса, логичнее заполнять страницу на уровне front-end используя Jquery и ajax. От spring mvc возьму только controller для вывода главной страницы.

**На веб-интерфейсе красота не требуется, можно просто в таблицу выводить всю информацию.**

Если есть возможность «дешево» продемонстрировать заказчику (а иногда и исполнителю) «демку» покрасивее и с дополнительными возможностями, то почему бы и нет? С помощью Jquery plugins, например, это просто и быстро сделать.

* + 1. Источник данных о курсах валют

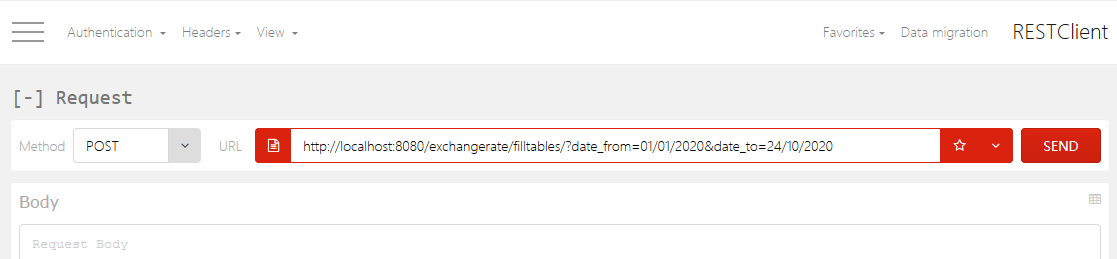
Для заполнения таблиц в БД о типах и курсах валют я использовал официальные данные Центробанка используя их REST интерфейсы:

<http://www.cbr.ru/scripts/XML_daily.asp?date_req=dd/mm/yyy>y

<http://www.cbr.ru/scripts/XML_dynamic.asp?date_req1=dd/mm/yyyy&date_req2=dd/mm/yyyy&VAL_NM_RQ=id>

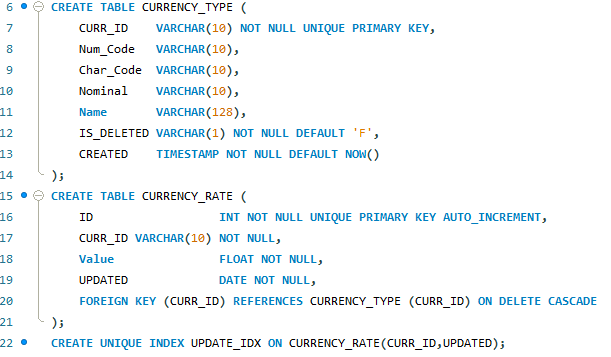
(id валюты берется из первого запроса)

Заполнить локальные таблицы в БД можно при помощи команды Веб-приложению:



Данные публикуются Центробанком не каждый день: в выходные и праздничные дни официальный курс не меняется. Поэтому необходимо помимо курса на текущую дату в Веб-интерфейсе приложения также отображать дату когда он был установлен.

* + 1. Структура базы данных

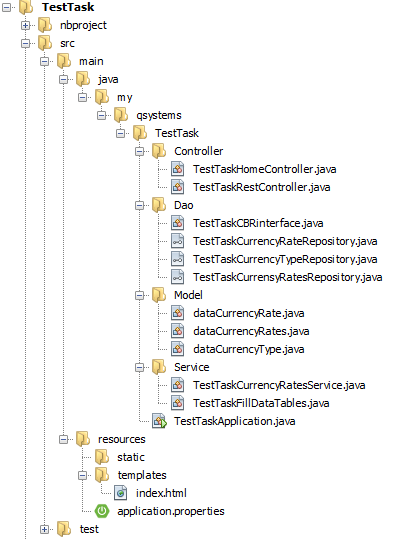


БД для Веб-приложению содержит две таблицы:

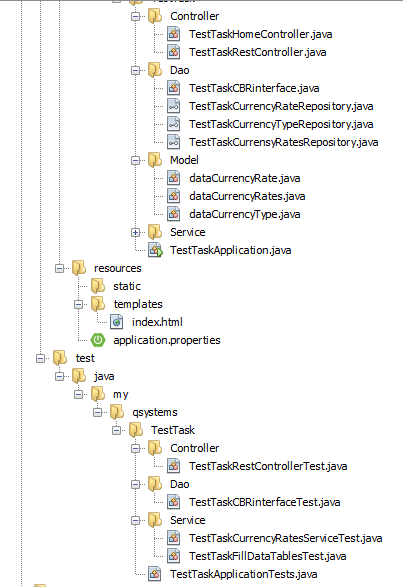
* CURRENCY\_TYPE – информация о типах валют
* CURRENCY\_RATE – о курсах валют на разные даты

Таблицы связаны между собой по ключу. Так как во второй таблице информации может быть много, на нее создается индекс, что позволяет минимизировать время выполнения запросов. Также создаются хранимые процедуры для выполнения необходимых запросов, чтобы максимально минимизировать использование SQL в Веб-приложении.

* + 1. Структура проекта



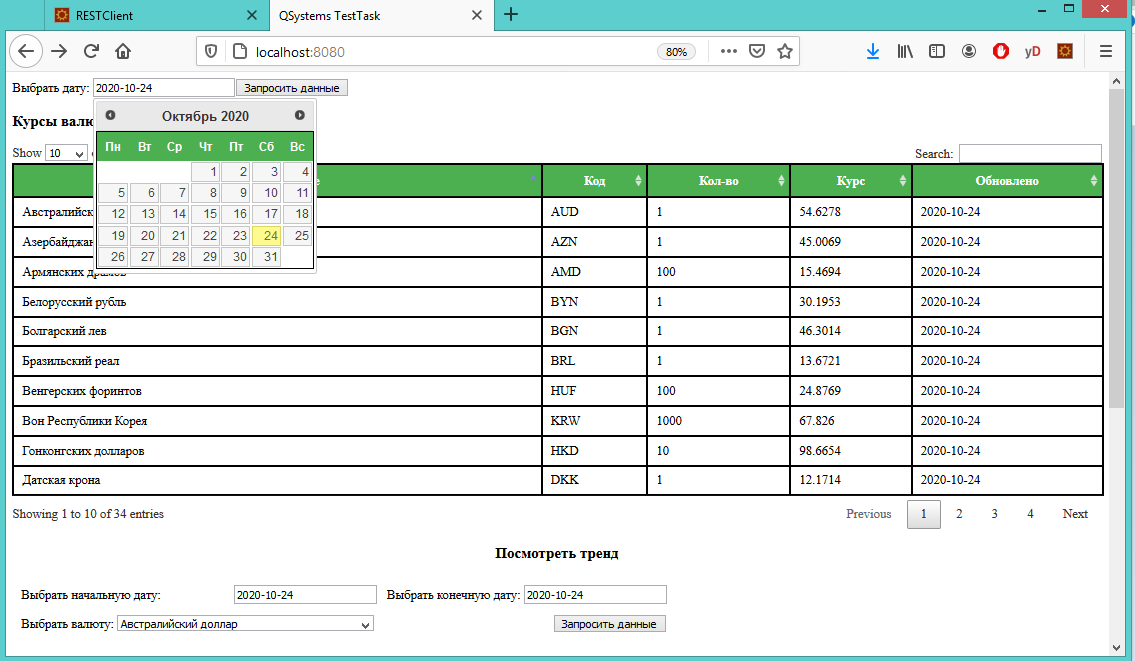
* Папка Controller – содержит mvc контроллер для отображения главной страницы (i**ndex.html**) и REST контроллер для доступа к данным и заполнению таблиц в БД
* Папка Dao – объекты для доступа к данным: локальной БД и данным центробанка. Классы **TestTaskCurrencyRateRenository** и **TestTaskCurrencyTypeRenository** – для заполнения таблиц в БД,  **TestTaskCurrencyRateRenository -** для чтения данных из таблиц с использованием SQL (т. к. запросы из объединенных таблиц), **TestTaskCBRinterface** – для чтения данных центробанка.
* Папка Model – классы определения структур данных
* Папка Service – классы содержащие функции используемые REST контроллером. **TestTaskFillDataTables** – служебный и не используется Веб-интерфейсом.
  + 1. Unit тесты

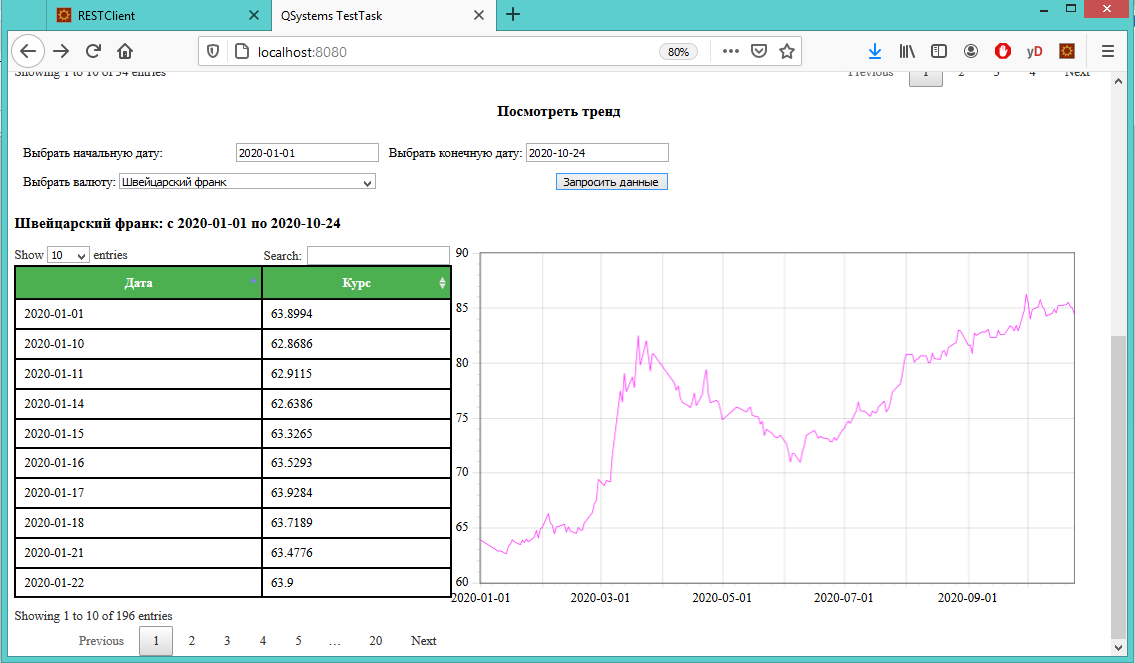


Unit тесты написаны лишь для некоторых классов которые что-то делают: нет смысла писать тесты для модельных классов и для классов которые которым не добавлено новых функций (например **TestTaskCurrencyRateRenository** не добавляет ничего к родным CRUD функциям, а лишь определяет тип данных с которыми работаем). Сложных тестов я писать не стал, т. к. в данной задаче и простых вызовов функций классов достаточно (если чего-то сломать, то простой тест это покажет).

* + 1. Пользовательский интерфейс

По умолчанию отображаются курсы всех доступных валют на текущий момент. Далее можно выбрать любую другую дату.

Также можно выбрать одну из валют и временной интервал и посмотреть и посмотреть динамику изменения курса.



Страница использует Jquery и ajax, поэтому нет при запросах не перегружается целиком.

* + 1. Средства разработки и исходники
* Java 11
* NetBeans IDE 12.0
* Jquery + plugins:
  + JqueryUI: <https://jqueryui.com/>
  + Datatables: <https://datatables.net/>
  + Flotcharts: [www.flotcharts.org](http://www.flotcharts.org/)

Исходники проекта доступны на GitHuh:

<https://github.com/dma1973/QStest>