

УДК 004.624

ИСПОЛЬЗОВАНИЕ LIQUIBASE  
ДЛЯ УПРАВЛЕНИЯ ИЗМЕНЕНИЯМИ СХЕМЫ БАЗЫ ДАННЫХ

Д.Ю. ДЁКИН

(Представлено: канд. физ.-мат. наук, доц. Ю.Ф. ПАСТУХОВ)

Рассматривается платформно-независимая библиотека с открытым исходным *Liquibase*, при помощи которой можно осуществлять контроль за изменениями схемы базы данных. Проанализированы некоторые возможности библиотеки *Liquibase*. Представлены базовые примеры, демонстрирующие простоту применения данной технологии в рамках управления изменениями структуры базы данных.

При разработке большинства приложений используются различные системы контроля версий (Git, Subversion и т.п.). Возникает вопрос, как с течением времени меняется схема БД, как это отслеживается, чем управляется? Случается, что изменения базы данных на тестовых и «боевых» серверах делаются вручную. Данный подход может не вызывать сложностей, если над проектом работает один или два разработчика, но, если работает целая команда, становится трудно синхронизировать все изменения между разработчиками.

**Liquibase** – платформно-независимая библиотека с открытым исходным кодом (реализована на Java), которая предназначена для управления изменениями схемы БД, а также является хорошим инструментом в миграции с одной базы данных на другую.

Чтобы начать работать с *Liquibase*, требуется указать несколько настроек для библиотеки:

1. Драйвер базы данных.
2. Путь к драйверу.
3. Url к базе данных.
4. Логин и пароль к базе данных.
5. Путь к чейнджлогу.

Основными понятиями, которыми оперирует *Liquibase*, являются:

1. Чейнджсет (от англ. changeset – набор изменений).
2. Чейнджлог (от англ. changelog – файл, состоящий из набора чейнджсетов).

Таким образом, отдельный чейнджлог представляет собой совокупность чейнджсетов в рамках какой-то отдельно взятой таблицы базы данных.

Код чейнджсетов может быть описан при помощи XML, JSON, YAML, SQL или даже на groovy. Например, при помощи следующего кода (листинг 1) можно создать таблицу **person** с двумя полями: *id*, *name*.

Листинг 1 – Создание таблицы «person»

```
<databasechangelog
  xmlns=http://www.liquibase.org/xml/ns/dbchangelog xmlns:xsi=http://www.w3.org/2001/XMLSchema-
instance xsi:schemalocation="http://www.liquibase.org/xml/ns/dbchangelog
http://www.liquibase.org/xml/ns/dbchangelog/dbchangelog-2.0.xsd">
  <changeset id="2017-09-09-00" author="dzmitry@mail.com">
    <createtable tablename="person">
      <column name="id" type="BIGINT" autoincrement="true">
        <constraints nullable="false" primaryKey="true"/>
      </column>
      <column name="name" type="VARCHAR(255)">
        <constraints nullable="false"/>
      </column>
    </createtable>
  </changeset>
</databasechangelog>
```

*Liquibase* имеет встроенную поддержку rollback (отмены) для некоторых типов чейнджсетов, как, например, для чейнджсета выше (*createTable*). Другие чейнджсеты автоматически быть откатенными не могут. В таких случаях предусмотрена секция *rollback*, позволяющая описывать действия для отмены изменений чейнджсета (листинг 2).

Листинг 2 – Пример чейнджсета с секцией «rollback»

```
<databasechangelog
  xmlns=http://www.liquibase.org/xml/ns/dbchangelog xmlns:xsi=http://www.w3.org/2001/XMLSchema-
instance xsi:schemalocation="http://www.liquibase.org/xml/ns/dbchangelog
http://www.liquibase.org/xml/ns/dbchangelog/dbchangelog-2.0.xsd">
  <changeset id="2017-09-09-00" author="dzmitry@mail.com">
    <insert tablename="person">
      <column name="id" value="John Smith"/>
    </insert>
    <rollback>
      DELETE FROM person WHERE name LIKE 'John Smith';
    </rollback>
  </changeset>
</databasechangelog>
```

Чейнджсеты в *Liquibase* определяются уникальностью комбинации тегов `id`, `author`, а также имени файла чейнджлога. Таким образом, чтобы успешно применять чейнджсеты (в рамках одного чейнджлога), необходимо задавать уникальное поле для `id` и `author`.

На практике же (чаще всего) в качестве `id` берется значение текущей даты, и в конец добавляется порядковый номер чейнджсета. Поэтому если бы в чейнджлог из листинга 2 нужно было бы добавить еще один чейнджсет, то его `id` имело бы значение «2017-09-09-01».

Имея несколько чейнджлогов, можно создать один кумулятивный файл, который будет импортировать другие чейнджлоги (листинг 3).

Листинг 3 – Пример кумулятивного чейнджлога

```
<databasechangelog
  xmlns=http://www.liquibase.org/xml/ns/dbchangelog>
  <include file="v-1.0/person-v-0-1.xml" relativeToChangelogFile="true"/>
  <include file="v-1.0/data-v-0-1.xml" relativeToChangelogFile="true"/>
</databasechangelog>
```

Используя *Liquibase*, стоит следовать ряду правил и рекомендаций, который облегчит управление версиями схемы базы данных в дальнейшем. Итак, версионность приложения должна быть отражена в структуре папок чейнджлогов:

1. Чейнджлоги изменения базы данных должны содержаться в отдельных папках, соответствующих версиям приложения
2. Держать кумулятивный файл чейнджлогов для каждой версии.
3. Соблюдать правила именования чейнджлогов.
4. Лучше много маленьких чейнджсетов, чем один большой.
5. Чейнджлоги, изменяющие схему базы данных, должны быть атомарными.

*Liquibase* оперирует значениями хеша чейнджсета. Поэтому если чейнджсет уже был применен, но оказалось, что нужно добавить что-то еще, то должен создаваться отдельный чейнджсет. Иначе, если отредактировать имеющийся чейнджсет, изменится хеш данного чейнджсета. В таком случае *Liquibase* не применит данные изменения к базе данных.

#### **Заключение**

Рассмотрены некоторые возможности библиотеки *Liquibase*, а также даны советы и рекомендации по организации и хранению чейнджсетов. Предоставлены базовые примеры, демонстрирующие простоту применения данной технологии в рамках управления изменениями структуры базы данных.

#### **ЛИТЕРАТУРА**

1. Хабрахабр [Электронный ресурс] Использование Liquibase без головной боли. 10 советов из опыта реальной разработки. – Режим доступа: <https://habrahabr.ru/post/179425>. – Дата доступа: 24.09.2017.
2. Хабрахабр [Электронный ресурс] Управление миграциями БД с Liquibase. – Режим доступа: <https://habrahabr.ru/post/179425>. – Дата доступа: 24.09.2017.
3. Ezcode [Электронный ресурс] Версионирование структуры БД с помощью Liquibase. – Режим доступа: <http://easy-code.ru/lesson/database-versioning-liquibase>. – Дата доступа: 24.09.2017.