MAYFLOWER

Agile Datenbankentwicklung mit Liquibase

Alberto Assmann I 07.12.11

Agile Softwareentwicklung

- Agile Methoden (Scrum, XP, Kanban) für Projektmanagement
- Continouos Improvments
- Code refactorings, pair programming
- Versionsverwaltung
- Testing Framework & Continouos Integration
- Buildautomatisierung

Und für die Datenbank?

- Keine Updateautomatisierung
- Keine genaue Versionierung
- Existiert meine Tabelle schon (IF EXISTS)?
- Fehlende Trennung zwischen Daten und Schema
- Datenverlust bei nachträglichen Änderungen

Liquibase

- Open Source (Apache 2.0)
- Plattformunabhängig durch Java
- XML zum beschreiben des Schemas
- Datenbankunabhängig

Unterstützte Datenbanken

H2

MySQL

ORACLE

Apache Derby

HSQL database engine

IBM DB2

PostgreSQL

Microsoft SQL Server

Informix

SYBASE An SAP Company

SQLite

SAP DB

InterSystems CACHÉ™ Make Applications Faster

Features

- Datenbankversionierung
- Updateautomatisierung
- Rollbacks im Fehlerfall (teils automatisch)
- Datenbankänderungsdokumentation
- Unterschiede zwischen Datenbanken ermitteln
- Taggen von Datenbanken
- XML Schema für Autovervollständigung

Changelogs & Changesets

- Jedes Skript ist ein <databaseChangeLog>
- Ein Changelog kann beliebig viele <changeSet>'s enthalten
- Innerhalb des Changesets werden die Änderungen in XML abgebildet
- Jedes Changeset muss eine eindeutige ID und einen Autor haben

```
<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="1" author="bob@mayflower.de">
                <createTable tableName="department">
                <column name="id" type="int">
                                <constraints primaryKey="true" nullable="false"/>
                </column>
                <column name="name" type="varchar(50)">
                                <constraints nullable="false"/>
                </column>
        <column name="active" type="boolean"
        defaultValueBoolean="true"/>
        </createTable>
   </changeSet>
</databaseChangeLog>
```

Datenbankversionierung - DATABASECHANGELOG

ID	AUTHOR	FILENAME	DATEEXECUTED	ORDEREXECUTED	EXECTYPE	MD5SUM	DESC
1	newloki	/Applications/MAMP/htdocs/schiffeversenken-server/...	2011-08-07 03:27:12	1	EXECUTED	3:dab57116ac459f99df68f41e644ab44d	Create
2	newloki	/Applications/MAMP/htdocs/schiffeversenken-server/...	2011-08-07 03:27:13	2	EXECUTED	3:c6ab72a32177c952ba745ec0892583b4	Create
3	newloki	/Applications/MAMP/htdocs/schiffeversenken-server/...	2011-08-07 03:27:13	3	EXECUTED	3:9593b05998f62de9f6c46e9d2a32abf6	Create

- DATABASECHANGELOG speichert Metainformationen
- Changesets werden bei Ausführung gehasht
- Reihenfolge, Autor, Ausführungsdatum und Hash werden gespeichert
- Finger weg von nachträglichen Änderungen am Changeset

Updateautomatisierung

- Integration in bestehende Deploymentprozesse
  - Kommandozeile
  - Ant
  - Spring
  - Grails
  - Maven
  - Phing als Erweiterung (siehe: https://github.com/bitExpert/liquibase-phing)
- Binary als JAR
- Treiber nicht Inklusive

```
java -jar liquibase.jar
    --driver=com.mysql.jdbc.Driver \
    --classpath=databasedriver/mysql-connector-java-5.1.17-bin.jar \
    --changeLogFile=../data/sql/changelog.xml \
    --url="jdbc:mysql://127.0.0.1:<MysqlPort>/<DB Name>" \
    --username=<UserName> \
    --password=<Password> \
    update
```

Rollback im Fehlerfall – Automatismen

- Automatischer Rollback für Anlegen und Umbennen
  - Create Table | View | Sequence | Index
  - Rename Table | View | Column
  - Add Lookup Table | Constraint | Column | Key

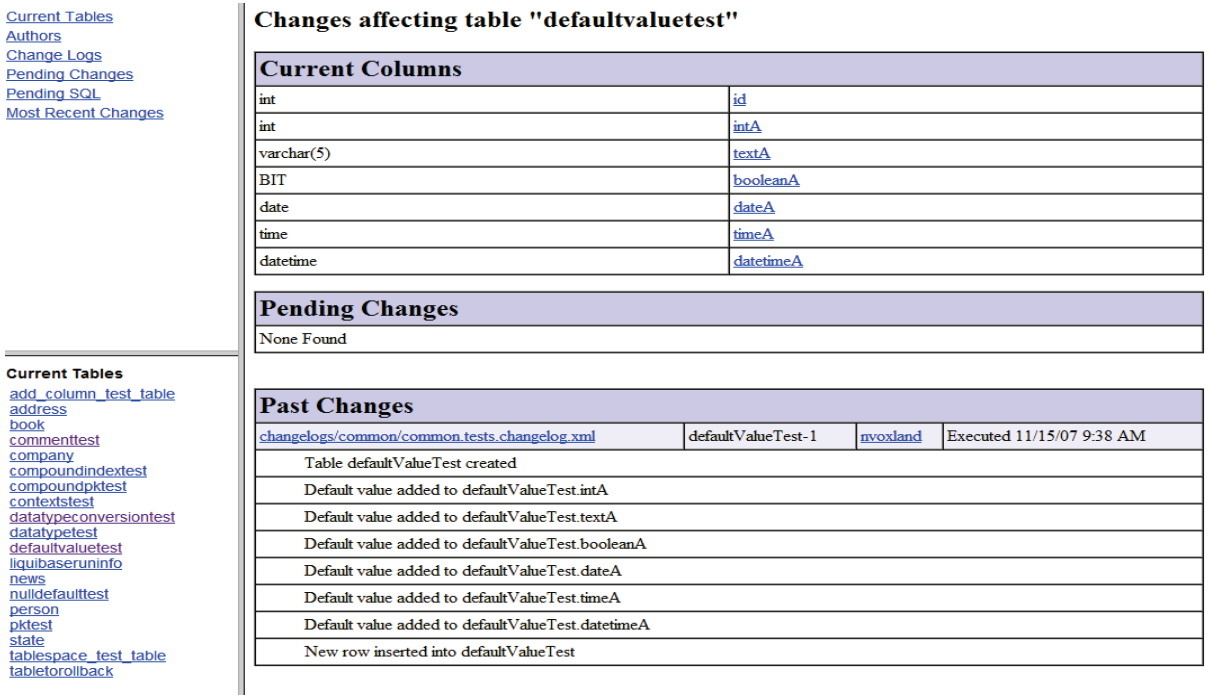
```
<changeSet author="alberto.assmann@mayflower.de" id="1">
   <dropTable tableName="i18n" />
   <rollback>
      <dropForeignKeyConstraint
            constraintName="fk_id_language"
                  baseTableName="i18n_languages"/>
<dropTable tableName="i18n" />
   </rollback>
</changeSet>
```

Rollback im Fehlerfall – von Hand

- Manueller Rollback
  - *rollback <tag>* zu Datenbank Tag
  - *rollbackCount <x>* x Changesets rückwärts
  - *RollbackToDate <date>* zu Datum

Änderungsdokumentation

- Nur per Kommandozeile möglich
- Ausgabe an Java-Doc angelehnt



Current Tables
Authors
Change Logs
Pending Changes
Pending SQL
Most Recent Changes

Current Tables
add_column_test_table
address
book
commenttest
company
compoundindextest
compoundpktest
contextstest
datatypeconversiontest
datatypetest
defaultvaluetest
liquibaseruninfo
news
nulldefaulttest
person
pktest
state
tablespace_test_table
tabletorollback

Changes affecting table "defaultvaluetest"

Current Columns	
int	id
int	intA
varchar(5)	textA
BIT	booleanA
date	dateA
time	timeA
datetime	datetimeA

Pending Changes
None Found

Past Changes			
changelogs/common/common.tests.changelog.xml	defaultValueTest-1	nvoxland	Executed 11/15/07 9:38 AM
Table defaultValueTest created			
Default value added to defaultValueTest.intA			
Default value added to defaultValueTest.textA			
Default value added to defaultValueTest.booleanA			
Default value added to defaultValueTest.dateA			
Default value added to defaultValueTest.timeA			
Default value added to defaultValueTest.datetimeA			
New row inserted into defaultValueTest			

Unterschiede zwischen Datenbanken

- *diff* – Änderungen zwischen zwei Datenbanken anzeigen
- *diffChangeLog* – Changelogs aus Änderungen generieren
- Nicht beachtet werden:
  - Stored Procedures
  - Check Constraints
  - Triggern
  - Functions

```
java -jar liquibase.jar
    [...]
    --changeLogFile=../data/sql/generatedChangelog.xml
    diffChangeLog
    --referenceUrl=<JDBC URL>
    --referenceUsername=<BENUTZER NAME>
    --referencePassword=<PASSWORT>
```

```
<changeSet author="albertoassmann (generated)" id="1318153136559-1">
    <createTable tableName="i18n">
        <column autoIncrement="true" name="id" type="BIGINT">
            <constraints nullable="false" primaryKey="true"/>
        </column>
        <column name="token" type="VARCHAR(255)">
            <constraints nullable="false"/>
        </column>
        <column name="default" type="TEXT">
            <constraints nullable="false"/>
        </column>
    </createTable>
</changeSet>
```

```
<changeSet id="1" author="alberto.assman@mayflower.de">
   <preConditions onFail="CONTINUE">
      <sqlCheck expectedResult="0">
         select count(*) from i18n where token = "hello"
      </sqlCheck>
   </preConditions>
   <insert tableName="i18n">
      <column name="token" value="hello"/>
      <column name="default" value="Hello World!"/>
   </insert>
</changeSet>
```

Bedingungen behandeln

- HALT, stoppen von Liquibase
- CONTINUE, beim nächsten mal ausführen
- MARK_RAN, Überspringen
- WARN, Warnung ausgeben und ausführen

```
<changeSet id="1" author="alberto.assmann@mayflower.de" context="test">
[...]
</changeSet>
```

```
java -jar liquibase.jar
    [...]
    --contexts=„test“
   update
```

Erfahrungswerte – First Steps

- Anfänglicher Lernaufwand
- Ein Experte im Team
- *generateChangeLog* für bestehende Datenbanken
- Immer erst lokal testen
- *--logLevel=debug*

Erfahrungswerte – Datenmigration

- Als PHP Skript
- Ausführung durch ***executeCommand***
- Skripts haben Rollbackoption
- Abhängigkeiten zu eigenen Backends/Entities/Models sind zu vermeiden

Erfahrungswerte – Einsatz im Team

- Unterteilung in einzelne Dateien
- Hauptdatei inkludiert die einzelnen Dateien
- Dateien werden erst nach abgeschlossener Entwicklung inkludiert (keine vorzeitigen Updates)
- Kein reines SQL mehr (auch wenn dies ausführbar ist)
- Bugfixes nicht mehr an Dateien die bereits inkludiert wurden

Vielen Dank für Ihre Aufmerksamkeit!

Kontakt

Alberto Assmann

alberto.assmann@mayflower.de

+49 931 35965 1164

Mayflower GmbH

Gneisenaustr. 10/11

97074 Würzburg

MAYFLOWER

Haben Sie noch Fragen?

webinar@mayflower.de
http://is.gd/webinar_liquibase

Folgetermin: Februar 2012