Liquibase Tutorial

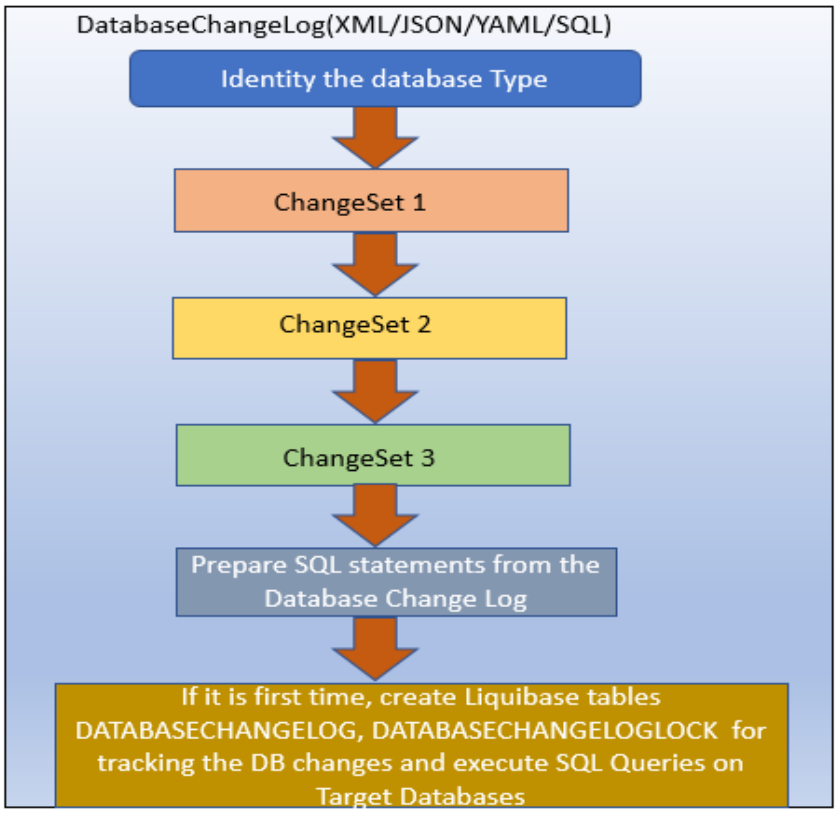
-By SivaReddy

1) What is Liquibase, why we need it and its advantages?

"Liquibase is an open source database-independent library for tracking, managing and applying database schema changes"

- Liquibase is Source Control tool for your database
- Supports code branching and merging
- Supports multiple developers
- Supports multiple database types**(mysql, postgresql, oracle, mssql, Sybase, asany, db2, derby, hsqldb, h2, Informix, firebird, sqlite)**
- Supports XML, YAML, JSON and SQL formats
- Supports context-dependent logic→ based on specific release version
- Cluster-safe database upgrades
- Generate Database change documentation
- Generate Database "diffs"
- Run through your build process, embedded in your application or on demand, E.g: you can integrate with ant/maven/gradle build tools
- Automatically generate SQL scripts for DBA code review (updateSQL, rollbackSQL)
- Does not require a live database connection
- Datical is both the largest contributor to the Liquibase project and the developer of Datical DB

2) How Does Liquibase Works?



Sample Database Changelog for xml:

```
<databaseChangeLog
xmlns="http://www.liquibase.org/xml/ns/dbchangelog"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:ext="http://www.liquibase.org/xml/ns/dbchangelog-ext"
xsi:schemaLocation="http://www.liquibase.org/xml/ns/dbchangelog
http://www.liquibase.org/xml/ns/dbchangelog/dbchangelog-3.0.xsd
        http://www.liquibase.org/xml/ns/dbchangelog-ext
http://www.liquibase.org/xml/ns/dbchangelog/dbchangelog-ext.xsd">

<preConditions>
<runningAsusername="sa"/>
</preConditions>

<changeSetid="1"author="Siva">
<createTabletableName="Customer">
<columnname="id"type="int"autoIncrement="true">
<constraintsprimaryKey="true"nullable="false"/>
</column>
<columnname="firstname"type="varchar(50)"/>
<columnname="lastname"type="varchar(50)">
<constraintsnullable="false"/>
</column>
```

```
<columnname="state"type="char(2)"/>
</createTable>
</changeSet>

</databaseChangeLog>
```

Sample Database Changelog for JSON:

```
{
    "databaseChangeLog": [
        {
            "preConditions": [
                {
                    "runningAs": {
                        "username": "sa"
                    }
                }
            ]
        },

        {
            "changeSet": {
                "id": "1",
                "author": "Siva",
                "changes": [
                    {
                        "createTable": {
                            "tableName": "Emp",
                            "columns": [
                                {
                                    "column": {
                                        "name": "id",
                                        "type": "int",
                                        "autoIncrement": true,
                                        "constraints": {
                                            "primaryKey": true,
                                            "nullable": false
                                        },
                                    }
                                },
                                {
                                    "column": {
                                        "name": "firstname",
                                        "type": "varchar(50)"
                                    }
                                },
                                {
                                    "column": {
                                        "name": "lastname",
                                        "type": "varchar(50)",
                                        "constraints": {
                                            "nullable": false
                                        },
                                    }
                                },
```

```
                                {
                                    "column": {
                                        "name": "state",
                                        "type": "char(2)"
                                    }
                                }
                            ]
                        }
                    }
                ]
            }
        },

        {
            "changeSet": {
                "id": "2",
                "author": "Siva",
                "changes": [
                    {
                        "addColumn": {
                            "tableName": "Emp",
                            "columns": [
                                {
                                    "column": {
                                        "name": "username",
                                        "type": "varchar(8)"
                                    }
                                }
                            ]
                        }
                    }
                ]
            }
        }

]

}
```

Sample Database Changelog for YAML:

```
databaseChangeLog:
  - changeSet:
      id: 1
      author: Siva
      changes:
        - createTable:
            tableName: Student
            columns:
              - column:
                  name: id
                  type: int
                  autoIncrement: true
                  constraints:
```

```
                    primaryKey: true
nullable: false
              - column:
                  name: first_name
                  type: varchar(255)
                  constraints:
nullable: false
              - column:
                  name: last_name
                  type: varchar(255)
                  constraints:
nullable: false
  - changeSet:
      id: 2
      author: Siva
      changes:
        - insert:
            tableName: Student
            columns:
              - column:
                  name: first_name
                  value: Siva
              - column:
                  name: last_name
                  value: Reddy
```

SpringBoot Liquibase Project:

1) **Add below dependencies in the pom.xml file**

```
<dependency>
	<groupId>org.springframework.boot</groupId>
	<artifactId>spring-boot-starter-jdbc</artifactId>
</dependency>
<dependency>
	<groupId>org.springframework.boot</groupId>
	<artifactId>spring-boot-starter-web</artifactId>
</dependency>
<dependency>
	<groupId>org.liquibase</groupId>
	<artifactId>liquibase-core</artifactId>
</dependency>

<dependency>
	<groupId>com.h2database</groupId>
	<artifactId>h2</artifactId>
	<scope>runtime</scope>
</dependency>

<dependency>
	<groupId>org.springframework.boot</groupId>
	<artifactId>spring-boot-starter-test</artifactId>
	<scope>test</scope>
```

```
</dependency>
```

2) To automatically run Liquibase database migrations on startup, add the **org.liquibase:liquibase-core** to your classpath.
3) By default, the master database change log is read from `db/changelog/db.changelog-master.yaml.` In addition to YAML, Liquibase also supports **JSON, XML, and SQL** change log formats

   If you want to refer database change logs from any other paths then add below property in application.properties file

```
spring.liquibase.change-log=classpath:/db/changelog/Create_Employee.json
spring.h2.console.enabled=true
management.endpoints.web.exposure.include=*
```

4) For the demo, we are using the H2 embedded database and springboot automatically identified the datasource based on the database added to the classpath. i.e database url, username and password
5) By default, Liquibase autowires the(`@Primary`) `DataSource` in your context and uses that for migration

   Use below mvn commands:

   mvn clean package
   mvn spring-boot:run

   To verify Liquibase migration DB scripts in the database,
   You can access the URL:http://localhost:8080/h2-console
   You can look at http://localhost:8080/actuator/liquibase to review the list of scripts.

   JDBC URL: **jdbc:h2:mem:testdb**
   Driver Class: org.h2.Driver
   User name: sa
   Password: empty (not required)