Development

Sentinel

# Einleitung

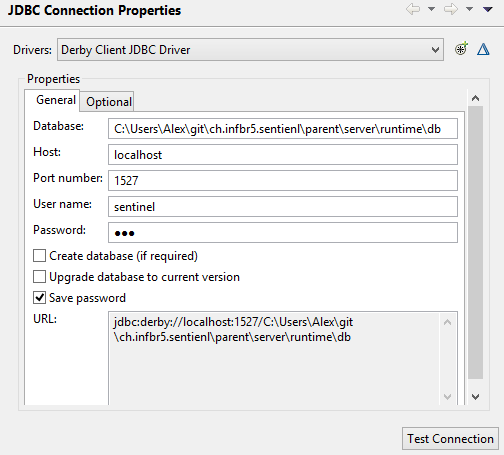
Dieses Dokument hält wichtige Design-Entscheidungen und soll den Entwickler unterstützen die Software Sentinel erfolgreich weiterzuentwickeln.

# Architektur

# Datenbank

Im Einsatz steht die Derby Datenbank, welche mit Hilfe von JPA (openjpa) zugegriffen wird.

Um die Datenbank anzuschauen, kann man in Eclipse (Database Development Tools) folgende Connection eintragen.



Zuvor muss man noch den Treiber konfigurieren. Der Treiber sollte bereits im Maven Repository heruntergeladen worden sein. Der Server muss in diesem Fall laufen, wenn man den Derby Client Driver wählt. Man verbindet sich zur aktuellen Instanz. (wenn man den embedded derby driver wählt, kann man sehr wahrscheinlich ohne laufenden Server die Instanz öffnen, jedoch kann dann der Server nicht mehr gestartet werden, bis die Insanz wieder zu ist.)

Das Passwort lautet aktuell PWD.

## Migrationen

Es ist kein eigentliches Schema definiert. Das überlassen wir JPA. JPA nimmt seine Entitäten und generiert das Schema. Bei Änderungen von Entitäten passt JPA das einfach an. Es kann aber jedoch sein, wenn man ein Attribut löscht, dass dann die Daten weg sind, obwohl man es vielleicht in einer anderen Identität haben möchte.

Aus diesem Grund wird hier erklärt wie man Migrationen durchführen kann. Dazu gibt es eine sehr gute Library, welche die grundlegenden Tasks übernimmt. Man muss nur noch die Migration schreiben.

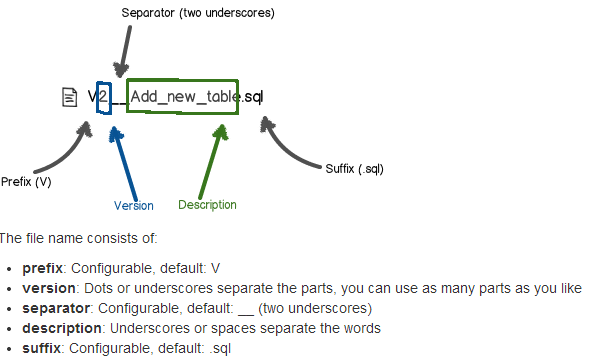
Die Library nennt sich flyway (<http://flywaydb.org/>). Diese Library wird aufgerufen bevor JPA zum Einsatz kommt und dann irgendwelche Anpassungen selber durchführen würde.

Flyway erstellt in der DB eine Tabelle namens „schema\_version“ für Sentinel habe ich diesen Wert auf „FLYWAY\_SCHEMA\_VERSION“ angepasst. Falls die Tabelle nicht vorhanden ist wird sie direkt erstellt. Dort sind alle Migrationen enthalten und auch den aktuellen Versionsstand der laufenden DB.

Nun zur eigentlichen Arbeit des Entwicklers, welche Daten migrieren möchte. Eine Migration kann man mittels SQL Scripts oder Java (JDBC) durchführen. Man muss lediglich eine gewisse Konvention einhalten.

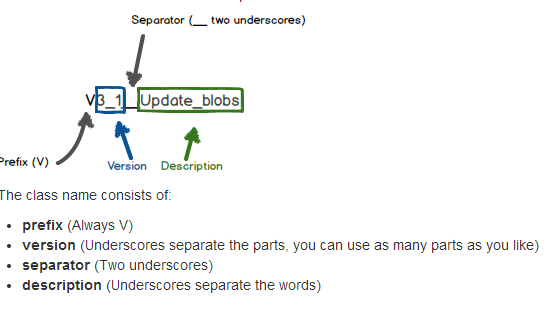
SQL Scripts:

Die SQL Scripts müssen unter src/main/resources/db/migration abgelegt werden. Und folgende Konvention einhalten:



http://flywaydb.org/documentation/migration/sql.html

Java-Klassen müssen unter /src/main/java/db.migration





<http://flywaydb.org/documentation/migration/java.html>

# Logging

Client wie Server besitzen drei Log4j-Appenders:

* ConsoleAppender
* FileAppender
* SystemMeldungAppender