RunSqlScriptHandler

Beschreibung

Mit dem RunSqlScriptHandler können Datenbankskripte (z.B. Sql queries) direkt aus dem Prozessmodell heraus ausgeführt werden. Ein Query-Ergebnis kann anschließend z.B. in der Smartform angezeigt werden. Das ist insbesondere sinnvoll, wenn Prozessteilnehmer Datenbankabfragen zur Bearbeitung der Tasks benötigen. Eine ausführliche Anleitung zum Einsatz des Handlers finden Sie im folgenden Beispiel.

Klasse

```
com.dooris.bpm.actionhandler.RunSqlScriptHandler
```

Event Type

beliebig

Action Name

beliebig

Mandatory Fields

leer

Parameter

databaseEngine (Default: oracle)

Gibt an, mit welcher Datenbank verbunden werden soll. Aktuell wird mysql, oracle und mssql als Wert unterstützt.

host

Der Parameter **host** enthält die URL des Hosts, auf dem die Datenbank läuft (z.B. host=get.taskinmotion.de;).

Dabei soll nicht die gesammte URL unter der die Datenbank zu erreichen ist hinterlegt werden, Ports

usw. werden mit anderen Parametern übergeben und TIM setzt die genaue Anfrage-URL selbst nach folgenden Schemata zusammen:

Im Falle von einer MySql Datenbank: “`jdbc:mysql:” + host + port + “/” + database“`

Im Falle einer Oracle Datenbank: “*jdbc:oracle:thin:@” + host + port + ”:“ + database”*

Im Falle einer MsSql Datenbank: “*Im Falle einer mssql Datenbank: “jdbc:sqlserver:” +* `host + port + ”;DatabaseName=“ + database”`

port

Hier wird der Port hinterlegt unter dem die Datenbank zu erreichen ist (z.B. port=17102;).

database

Hier wird der Name der Datenbank hinterlegt, die man erreichen will (z.B. database=EmployeeDb;).

user

Hier wird der Benutzername für die Zugriffskontrolle hinterlegt

pass

Hier wird das Passwort für die Zugriffskontrolle hinterlegt

query

Der **query** parameter enthält die SQL Query, die auf der Datenbank ausgeführt werden soll. (z.B. query=SELECT * FROM Mitarbeiter WHERE Gehalt>50000;)

Wenn im query auf Prozessvariablen zugegriffen werden soll, kann dies mit '${variable}' erfolgen (z.B. query=SELECT * FROM Mitarbeiter WHERE Gehalt>'${variable}';)

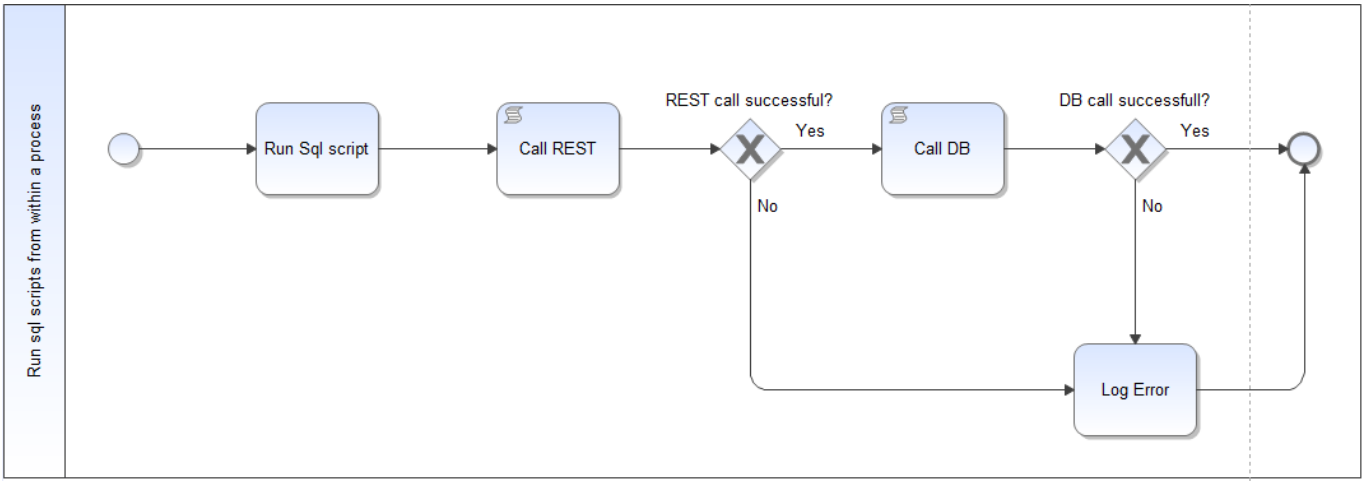
Der RunSqlScriptHandler gibt **natives SQL** weiter! Dadurch kann das gesamte Spektrum an **SQL Befehlen** über den Handler abgewickelt werden (z.B. Datenbanken erstellen bzw. löschen, Tabellen anlegen, bearbeiten und löschen usw.)

responseVariable

Im Parameter **responseVariable** wird der Name der Variable hinterlegt, in die das query Ergebnis gespeichert wird. Mit der Variable kann das query Ergebnis z.B. in der Smartform angezeigt werden.

Beispiel

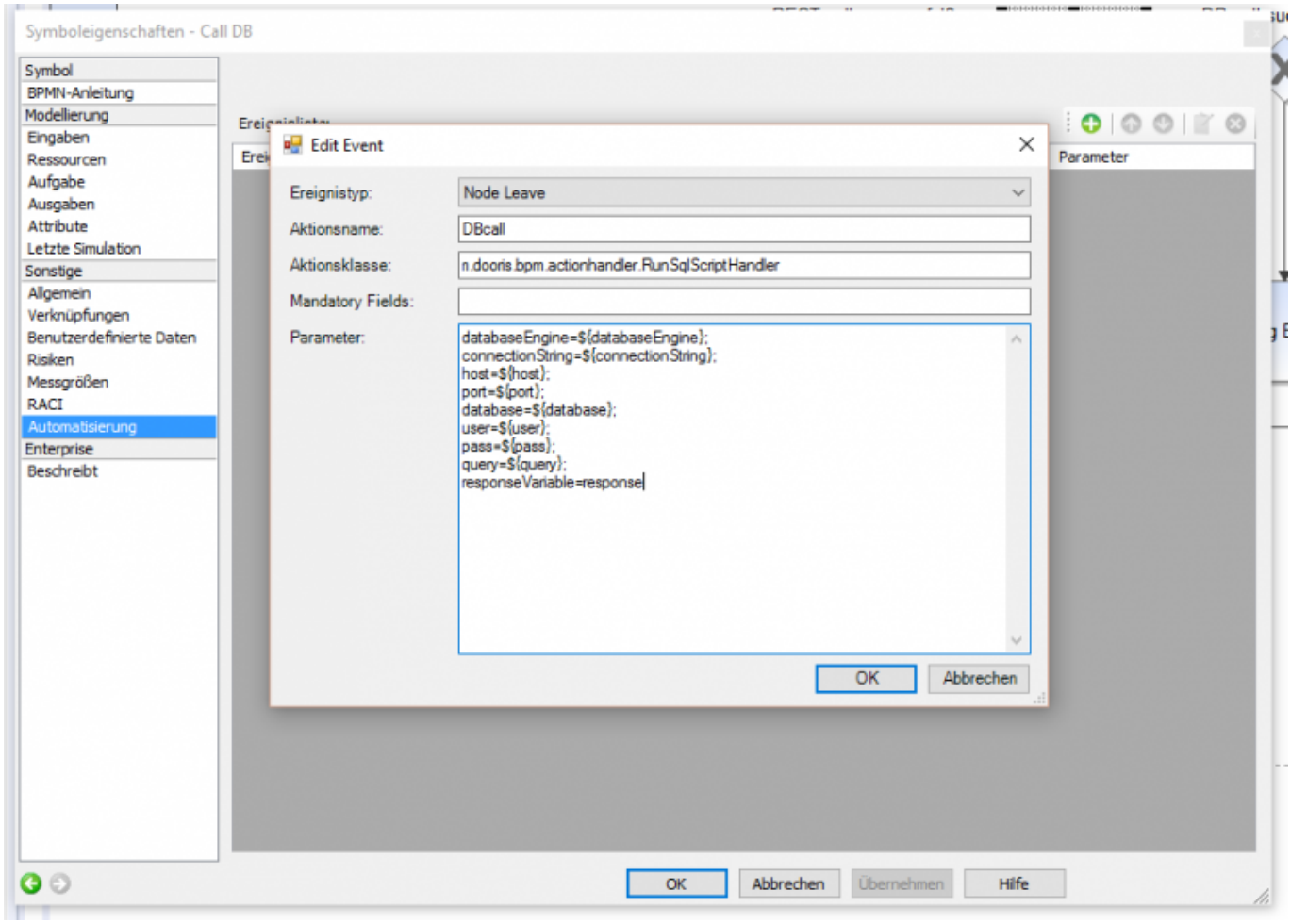
Das folgende Beispiel illustriert eine Möglichkeit, wie ein Sql Skript direkt aus einem Prozess heraus ausgeführt werden kann. Dabei werden die Parameter über die Smartform aufgenommen und als Prozessvariablen an den RunSqlScriptHandler weitergegeben. Das Prozessmodell sieht folgendermaßen aus.



Die benötigten Parameter werden in der Aktivität **Run Sql script** erfasst. Nach dem Ende der Aktivität wird über die Skript Aktivität **Call REST** ein REST call mithilfe des HttpRestHandler gestartet. Ist dieser erfolgreich wird die eigentliche Sql query in der Skript Aktivität **Call DB** ausgeführt. Im Fehlerfall wird der Fehler in eine Log-Datei geschrieben.

Der RunSqlScriptHandler

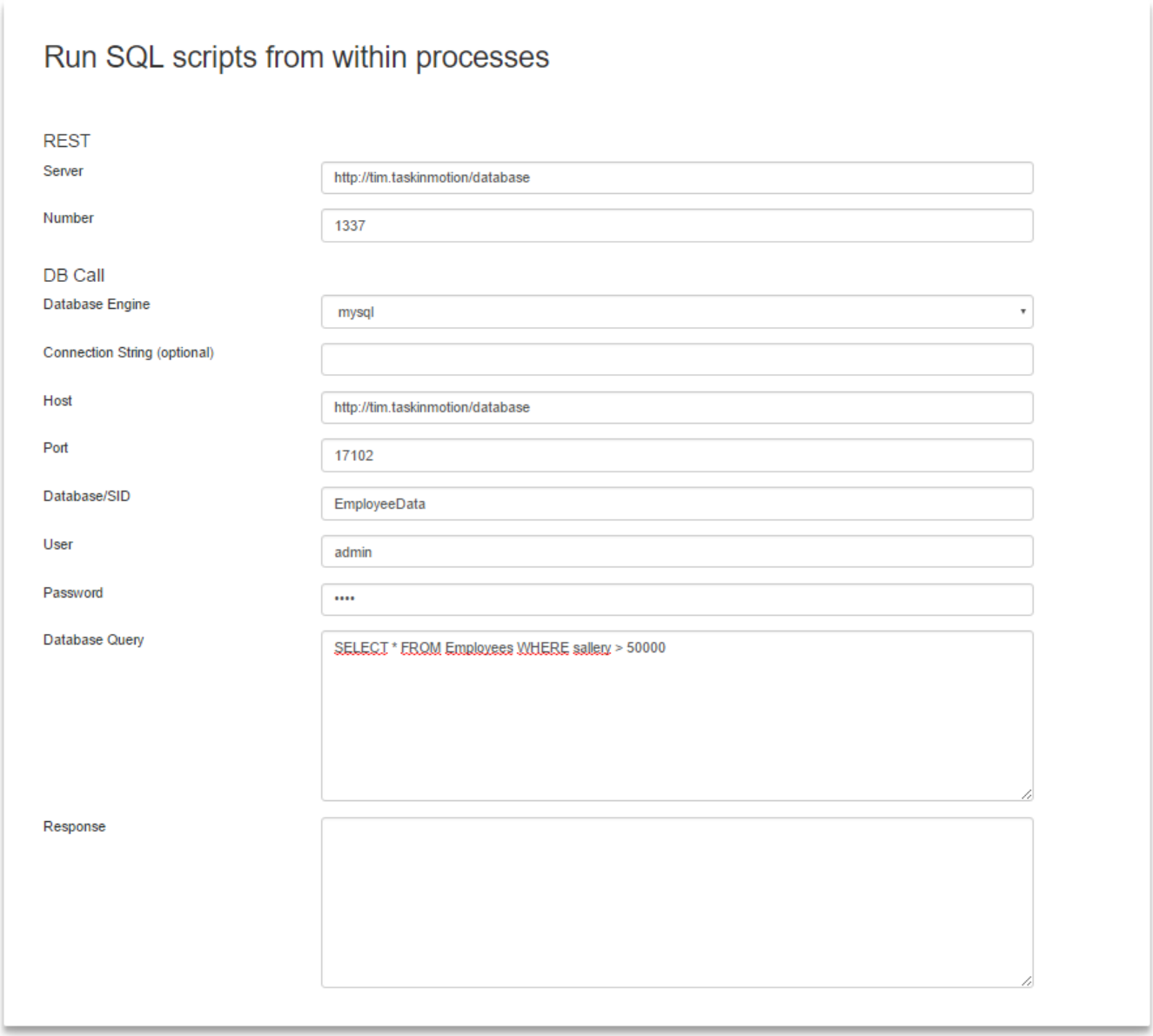
Der RunSqlScriptHandler befindet sich auf der **Call DB** Aktivität und erwartet die obigen Parameter. Im Zuge der Aktivität **Run Sql script** werden die nötigen Eingaben über die Smartform aufgenommen und als Prozessvariablen für den RunSqlScriptHandler und den HttpRestHandler bereitgestellt (s. Screenshot).



Die Smartform

Die Smartform dient lediglich der Aufnahme der benötigten Parameter. Zur Veranschaulichung werden in diesem Beispiel alle Parameter als Prozessvariablen zur Verfügung gestellt. In anderen Fall kann es sinnvoller sein lediglich den Benutzer, das Passwort sowie die Sql query über die Smartform abzufragen. Zu beachten ist auch, dass die Parameter **Server** und **Number** für den **REST call** nicht aber für den **DB call** benötigt werden. Das Ergebnis des DB calls kann im Textfeld **Response** ausgegeben werden.

Wie oben angedeutet setzt der RunSqlScriptHandler natives Sql ab. Über die Query können Datenbanken und Tabellen angelegt, bearbeitet und gelöscht werden. Neben dem Auslesen von Daten können Prozessdaten in die Datenbank eingebracht werden.



From:
https://wiki.tim-solutions.de/ - **TIM Wiki / NEW TIM 6 Documentation**

Permanent link:
https://wiki.tim-solutions.de/doku.php?id=software:tim:actionhandler:runsqlscripthandler&rev=1550485048

Last update: **2021/07/01 09:58**