

Manueller Ruleset Start

Version : 7.00.027

Normalerweise führt der ELOas die definierten Rulesets intervallgesteuert aus. Es gibt aber auch Vorgänge, die in der Abarbeitung so umfangreich sind, dass sie nicht in kurzen Intervallen ausgeführt werden können, auf der anderen Seite aber auch möglichst schnell nach einer bestimmten Veränderung aktiv werden müssen. Hierzu gibt es die Möglichkeit, dass die Ausführung eines Ruleset durch eine URL manuell (bzw. per Skript) gestartet wird.

Inhalt

1	Beispiel	2
2	Aktivierung des Ruleset	3
3	Weitere Hinweise	5
3.1	Asynchrone Triggerung des Ruleset	5
3.2	Berechtigungsprüfung	6
3.3	Ausführungsreihenfolge	6

1 Beispiel

Das nachfolgende Beispiel zeigt, wie man aus einem Client Skript heraus einen Ruleset aufrufen kann, der dann bestimmte Objekte verändert.

Achtung: da der Aufruf über einen http Zugriff erfolgt, kann prinzipiell jeder Anwender auch per Browser oder per Skript Befehl diese Aktion auslösen. Es muss deshalb sicher gestellt werden, dass die Funktion nicht missbraucht werden kann (z.B. durch Kontrolle der Anwendernummer oder durch eine feste interne Vorgabe der Objekt IDs).

Zuerst soll der verwendete Ruleset betrachtet werden. Durch die Angabe eines Intervalls von null Minuten (<interval>0H</interval>) wird dieser Ruleset als manuell getriggert definiert. Er wird also nicht zyklisch aufgerufen sondern wartet auf den Empfang einer bestimmten URL.

```
<ruleset>
  <base>
    <name>Expand Name</name>
    <search>
      <name>"OBJIDS"</name>
      <value></value>
      <mask>2</mask>
      <max>200</max>
    </search>
    <interval>0H</interval>
  </base>
  <rule>
    <name>Expand Name</name>
    <condition></condition>
    <script>
      log.debug("Param1: " + EM_PARAM1);
      log.debug("UserId: " + EM_USERID);
      NAME = "Freigegeben: " + NAME;
      EM_WRITE_CHANGED = true;
    </script>
  </rule>
  <rule>
    <name>Global Error Rule</name>
    <condition>OnError</condition>
    <script></script>
```

```
</rule>  
</ruleset>
```

Der interessante Teil liegt im Skript Bereich:

```
<script>  
  log.debug("Param1: " + EM_PARAM1);  
  log.debug("UserId: " + EM_USERID);
```

Der Aufruf kann bis zu drei Parameter mitgeben, diese können vom Ruleset über die Variablen EM_PARAM1, EM_PARAM2 und EM_PARAM3 abgefragt werden. Zudem kann das Skript optional das Ticket der aktuellen Anmeldung zur Authentifizierung mitgeben. In diesem Fall ist die Variable EM_USERID mit der Nummer des angemeldeten Anwenders gefüllt. Wenn keine Authentifizierung vorliegt, ist die Anwendernummer mit -1 belegt. Im ersten Parameter können eine oder mehrere Objekt-Ids übergeben werden, diese überschreiben dann den Search-Value aus der Ruleset Definition. Als Indexzeilenname muss in diesem Fall OBJIDS angegeben werden.

```
NAME = "Freigegeben: " + NAME;
```

Im Beispiel wird der Kurzbezeichnung der ausgewählten Objekte einfach der Text „Freigegeben“ vorangestellt. Hier können aber auch beliebige andere Veränderungen des Sord Objekts durchgeführt werden.

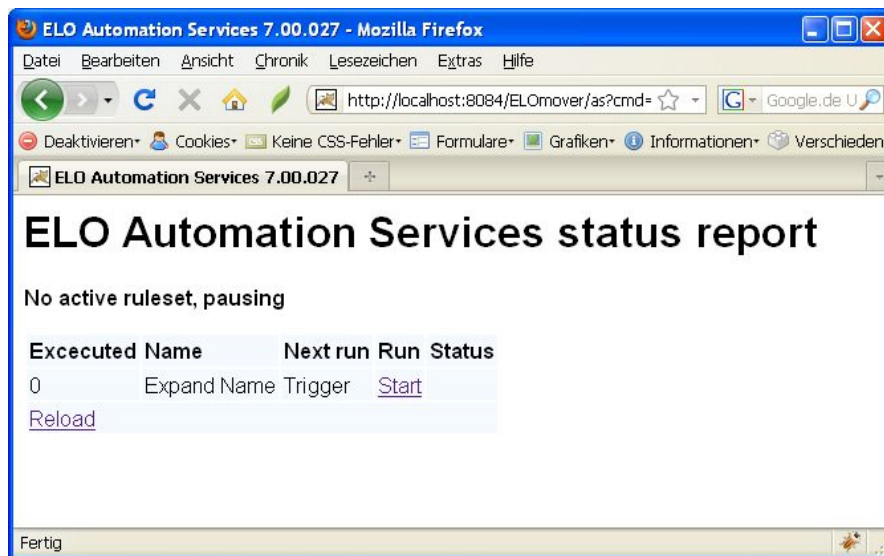
```
EM_WRITE_CHANGED = true;
```

Da das Objekt verändert wurde, soll es auch gespeichert werden.

```
</script>
```

2 Aktivierung des Ruleset

Nach dem Start des ELOas wird dieser Ruleset mit gestartet aber noch nicht aktiv. Er wartet auf einen externen Trigger (sichtbar an dem Text „Trigger“ im Feld „Next run“).



Die Triggerung kann durch den Aufruf einer URL, oder aus dem Windows Client heraus durch einen Skript Befehl erfolgen (ab Client Version 7.00.056):

SendELOasRequest(<Servername>, <Portnummer>, <Servicename>, <mit Ticket>, <Ruleset Name>, <Parameter 1>, <Parameter 2>, <Parameter 3>)

Servername	Name oder IP Adresse des ELOas Servers.
Portnummer	Portnummer des ELOas Servers. Im Normalfall 8080, Standard http Port.
Servicename	Servicename des ELOas Servers. In einer Standardinstallation setzt er sich aus dem Präfix as- und dem Archivnamen zusammen (z.B. as-ELO). Dabei ist unbedingt auf die korrekte Groß/ Kleinschreibweise zu achten, andernfalls meldet der Tomcat Server einen Fehler.
Mit Ticket	<p>0: keine Anmeldeinformation mit senden</p> <p>1: aktuelles Ticket als Anmeldeinformation mit senden. In diesem Fall prüft der ELOas das Ticket und ermittelt hieraus die Anwendernummer. Diese Information wird dann dem Ruleset zur Verfügung gestellt. Im Ruleset kann dann entschieden werden, ob und in welchen Umfang die Aktion ausgeführt wird.</p> <p>Hinweis: Im Augenblick kann die Anmeldeinformation bei der SSO Anmeldung nicht ausgewertet werden. Das wird in der nächsten Indexserverversion geändert.</p>
Ruleset Name	Name des auszuführenden Ruleset. Es können nur getriggerte Rulesets aufgerufen werden. Bei intervallgesteuerten Rulesets wird der Aufruf ignoriert.

Parameter1	Erster Parameter. Dieser Parameter wird, wenn er nicht leer ist, als Suchbegriff für die Ruleset Ausführung verwendet.
Parameter2 Parameter3	Weitere optionale Parameter. Diese können vom Ruleset abgefragt werden und die Ausführung steuern.

Das komplette Beispielskript für einen Aufruf kann dann so aussehen. Es ruft den Ruleset „Expand Name“ für die Objekte mit der ObjId 7944 und 7945 auf.

```
Set Elo=CreateObject("Elo.Professional")
MsgBox Elo.SendELOasRequest("localhost", 8084, "/ELOmover/as" , 1, "Expand Name",
"7944,7945", "TestParam2", "")
```

Der Ruleset kann auch aus beliebigen anderen Applikationen durch Aufruf einer URL getriggert werden:

<http://localhost:8084/ELOmover/as?cmd=run&name=Expand%20Name¶m1=7944,7945¶m2=TestParam2>

Beachten Sie bitte, dass in diesem Fall keine Authentifizierungsinformation mitgegeben werden kann. Sie müssen also in Ihrem Ruleset sicher stellen, dass kein Missbrauch erfolgen kann.

3 Weitere Hinweise

3.1 Asynchrone Triggerung des Ruleset

Wenn ein Ruleset durch eine URL oder einen Skriptaufruf getriggert wird, führt der ELOas diesen asynchron aus. Falls also gerade ein anderer Ruleset aktiv ist, wird die Skriptaussführung nicht so lange verzögert, bis der ELOas verfügbar ist. Statt dessen wird der Aktivierungsbefehl in eine Warteschlange gestellt und bei der nächsten Möglichkeit ausgeführt.

Das hat zwei Konsequenzen: das Client Skript kann sich nicht darauf verlassen, dass mit der Abarbeitung des Befehls tatsächlich auch die Operation durchgeführt wurde. Falls das für den weiteren Skriptverlauf wichtig ist, muss das im Skript selber geprüft werden und in eine Warteschleife integriert werden. Beachten Sie dabei jedoch, dass der ELOas möglicherweise gerade sehr umfangreiche andere Aktionen ausführt. Im Allgemeinen sollte ein Skript also nicht auf den Abschluss einer ELOas Aktion warten.

Weiterhin kann es sein, dass ein ungeduldiger Benutzer die Triggerung mehrfach veranlasst. In diesem Fall wird der Ruleset auch mehrfach ausgeführt. Er sollte also so angelegt sein, dass

diese mehrfache Triggerung nicht zu Fehlern führt, z.B. indem das Objekt vorab geprüft wird und die wiederholte Ausführung dann abgebrochen wird.

Noch eine Folge entsteht aus der asynchronen Ausführung: Fehler, die beim Abarbeiten des Rulesets aufgetreten sind, können nicht über den Aufruf zurück gemeldet werden.

3.2 Berechtigungsprüfung

Beim Aufruf aus dem Windows Client kann optional das Client Ticket der Anmeldung mit übertragen werden. In diesem Fall kann der ELOas die Anmeldung prüfen und den aktuellen Anwender ermitteln. Bei kritischen Aktionen sollte im Ruleset auf jeden Fall eine Prüfung auf einen berechtigten Anwender erfolgen und bei fehlender oder unzureichender Anmeldung eine Ausführung abgebrochen werden.

Eine anonyme Triggerung kann in bestimmten Fällen durchaus akzeptabel sein. Z.B. weil ein bestimmtes festgelegtes Objekt bearbeitet wird. In diesem Fall muss man darauf achten, dass die ObjektId nicht durch den Aufruf verändert werden kann. Das kann am einfachsten im onstart Event erfolgen indem der Wert von EM_SEARCHVALUE im Skript gesetzt wird. In diesem Fall wird für die Suche nicht der Parameter sondern der voreinstellte Wert aus dem Ruleset Skript verwendet.

3.3 Ausführungsreihenfolge

Ein manuell getriggertes Ruleset fügt sich ganz normal in die Ausführungsreihenfolge der Rulesets ein. Wenn zu einem Ruleset mehrere Trigger vorliegen, werden erst alle Trigger abgearbeitet bevor der nächste Ruleset bearbeitet wird.