SignalByMailReply

Beschreibung

Dieser Timer prüft Postfächer und schickt Instanzen weiter, wenn dort Antwort-Emails sind, welche Instanzen weiter signallen können. Er muss mit einem VariableDecisionHandler auf einer Node benutzt werden, wenn nach dem Signal eine Entscheidung getroffen werden soll. Der Absender wird, falls der Signal erfolgreich ist, als Actor auf allen Tasks der aktuellen Node gesetzt. Des Weiteren werden alle eingegangenen Mails, wenn Sie einer passenden Aktivität zugeordnet werden können, als Notiz an die Aktivität angehängt. Ein mit RegExp bestimmbarer Teil der erfolgreichen Mail wird als Variable (mit dem Namen *Nodename*-answer_successful) im Prozess gesetzt.
Im Mailtext muss die betreffende Prozessinstanz in der Form

```
-----(${SYS.PROCESSINSTANCE_ID})-----
```

angegeben werden. Die Antwort kann in der ersten Zeile, oder wahlweise in der Form

```
-----[Antwort: ]-----
```

angeben werden.

Webservice Name

```
ProcessInstanceManager
```

Webservice Methode

```
signalByMailReply
```

Parameter

alte Version:
email,password,port,host,limit,moveFolder,piIdRegExp,pDName,NodeNameRegExp,firstline,box,decisionRegExp,signalSuccessMail,processVariable,transitions
neue Version:
username,password,port,host,moveFolder,pDName,NodeNameRegExp,signalSuccessMail,processVariable,transitions,decisionRegExp

1. email

Email, die komplette Email-Adresse von der die Mails abgerufen werden sollen.

2. password

Password, das Passwort des Email-Accounts.

3. port

Port, passend zur Email-Adresse und dem Host. Häufige Ports: 110 *Pop* 995 *Pop mit Verschlüsselung* 143 *IMAP* 993 *IMAP mit Verschlüsselung*

4. host

Host, z.B. pop3.gmail.com oder imap.gmail.com (Meist etwas mit pop, pop3 oder imap. smtp ist normalerweise falsch.)

Wichtig: Sollen Mails in andere Ordner verschoben werden, so ist ein IMAP Postfach notwendig. Grundsätzlich hängt der Einsatz des Protokolls vom Postfach ab. Sollten beide Protokolle unterstützt werden, so ist IMAP zu bevorzugen.
Grundsätzlich sollte pro Timer und Prozessdefinition ein separates Postfach verwendet werden, um eine konsistente Funktionsweise des Timer gewährleisten zu können.

5. moveFolder

Kein Pflichtfeld, wenn gesetzt: Der Ordnername in den die erfolgreich ausgelesen (und zum signal benutzten) Mails verschoben werden sollen.

6. pDName

Name der Prozessdefinition.

7. nodeNameRegExp

NodeNameRegExp ist eine Regular Expression, die angibt wie der Nodename gefunden wird. Es wird im Mailtext und im Betreff danach gesucht. z.B. ++(.+)++ Dabei ist das gesuchte Element hier die der NodeName in Klammern zu schreiben. Beispiel:

```
++Wait++
```

Erklärung siehe auch piIdRegExp.

8. signalSuccessMail

Bei *true* wird auch bei erfolgreichem Signal eine Antwortmail an den Absender geschickt. Die Antwort kann mit Variablen im Prozess konfiguriert werden. Siehe unten.

9. processVariable

Der Wert der Prozessvariablen wird auf die gewählte Transition gesetzt, damit der VariableDecisionHandler im XOR-Gateway die Entscheidung treffen kann.

10. leavingTransitions

Wie wird der Prozessweg gefunden, wenn nach der aktuellen Aktivität mehrer Aktivitäten folgen können? Transitionen werden im Timer als Parameter mit angegeben - mit Mapping, welche Begriffe alle für diese Transition gelten. Nur eine Transition muss auch funktionieren - auch hier muss einer der korrekten Begriffe in der Mail stehen. Beispiel: freigabe:ok-okay-ja-j-freigegeben-freigabe-akzeptiert-von mir aus-angenommen-i.o.-i. o.- io-in ordnung-i. ordnung-in o.;abgelehnt:nok-nichtok-nicht okay-nein-n

Die Transitionen der Node sind: 'freigabe' und 'abgelehnt'. Für 'freigabe' z.B. kann auf die Mail mit 'ok', 'okay', usw. geantwortet werden. Für 'abgelehnt' sind es 'nok', 'nichtok', 'nicht okay', 'nein' und 'n'.

11. reasonRegExp

Die Entscheidung in der Mail, die den Prozess gesignaled hat, soll für den weiteren Prozessverlauf gespeichert werden. Hier kann angegeben werden wie diese im Mailtext gesucht wird. Wenn die Entscheidung ein Ergebnis, das über mehrere Zeilen geht, liefern soll dann muss die RegExp mit ?s enden. z.B. (.*?)Antwortet:?s

---

Variablen

Die Antworten auf ungültige Nachrichten können per Variablen oder in der loom.properties gesetzt werden:
no-tim-user-found-for-signal-by-mail-reply
user-not-actor-for-signal-by-mail-reply
found-no-transition-for-signal-by-mail-reply
found-no-node-for-signal-by-mail-reply
signal-done-for-signal-by-mail-reply
*found-no-processinstance-for-signal-by-mail-reply*Hier ist nur die Konfiguration über die loom.properties möglich.
Die Korrekte Antwort wird unter
signal-successful-signal-by-mail-reply
konfiguriert.

Auch der Name des abzurufenden Ordners kann in der loom.properties unter *inbox-name-signal-by-*

mail-reply konfiguriert werden.

> Achtung!
>
> > Es können in diesen Antworten je nach Schritt folgende Variablen benutzt werden:
> > *emailFrom*, ist der Absender
> > *userNameLast*, Nachname des Tim-Users
> > *userNameFirst*, Vorname des Tim-Users
> > *nodeName*, Name des zu signalden Node
> > *nodeEndDate*, Zeitpunkt zu dem die Node beendet wurde. Nur bei Signal-versuchen die nach Abschluss der Node versucht werden
> > *nodeEndTime*, Zeitpunkt zu dem die Node beendet wurde. Nur bei Signal-versuchen die nach Abschluss der Node versucht werden
> > *lastUserNameLast*, Nachname des Users der die Node beendet hat. Nur bei Signal-versuchen die nach Abschluss der Node versucht werden
> > *lastUserNameFirst*, Vorname des Users der die Node beendet hat. Nur bei Signal-versuchen die nach Abschluss der Node versucht werden

Nach dem Signal sind folgende Variablen verfügbar: *nodename*-lastActor *nodename*-signalDateTime (Format: „yyyy-MM-dd HH:mm:ss“)

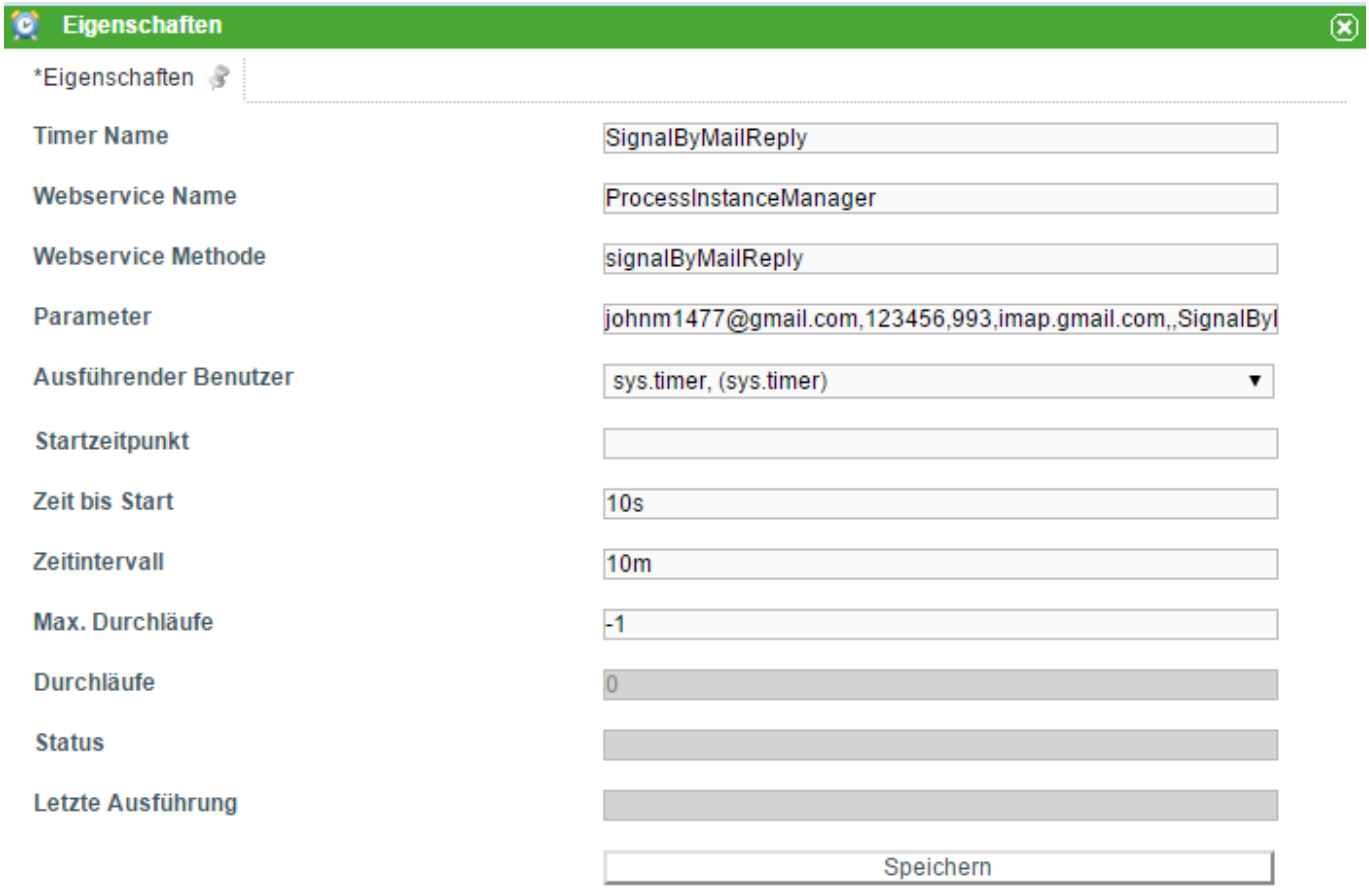
Ablauf

Zuerst wird das Postfach geöffnet, welches im Timer hinterlegt wurde, und die Mails abgeholt. Standardmäßig wird der Ordner *Inbox* abgeholt, dies kann aber in den loom.properties mit dem Parameter *inbox-name-signal-by-mail-reply* konfiguriert werden.
Danach werden die ersten 5 gefunden Mails abgearbeitet. Die Anzahl kann über den Parameter *inbox-name-signal-by-mail-reply* in den loom.properties konfiguriert werden. In jeder Mail wird zuerst im Inhalt und dann im Betreff nach einem Text in der Form wie —-(1234)—- gesucht. Dabei handelt es sich um die TIM-Prozessinstanz-ID.
Lässt sich keine Prozessinstanz-ID oder zu der ID keine Prozessinstanz in TIM finden, wird die Nachricht ignoriert.
Im nächsten Schritt wird mit der Absende-Email ein TIM-User gesucht. Sollte kein User gefunden werden, erhält der Absender eine entsprechende Mail.
Nun sucht der Timer mit der NodeNameRegExp nach dem Namen einer Aktivität im Mailtext und darauf im Betreff. Ist ein Name gefunden worden wird geprüft, ob die gefundene Prozessinstanz von der übergebenen Prozessdefinition stammt. Sollte dies nicht der Fall sein war die Mail nicht für diesen Timer bestimmt und auch diese Nachricht wird ignoriert. Es gibt keine Antwortmail.
Sollte dieser Timer für die Mail zuständig sein, wird geprüft, ob der Prozess an der richtigen Aktivität steht. Ist dies der Fall, wird die Mail als Notiz an die Aktivität angehängt. Wenn der Prozess schon weitergeleitet wurde bekommt der Absender eine entsprechende Nachricht.
Jetzt wird geprüft, ob der TIM-User der die E-Mail geschickt hat, für eine der Tasks auf der aktuellen Aktivität zuständig ist. Ist das nicht der Fall wird er informiert.
Nun wird geprüft, ob es mehrere Folge-Aktivitäten gibt. Sollte nur eine vorhanden sein, wird der User als Actor auf die Aufgaben der Aktivität gesetzt. Gibt es mehrere Möglichkeiten den Prozess fortzusetzen wird geprüft, ob die Mail eine Entscheidung dazu enthält.

Dann wird auch in diesem Fall der User Actor auf den Task. Jetzt werden Prozessvariablen gesetzt und der Prozess weiter geschickt. Falls gewünscht wird auch dann eine Antwort verschickt.
Alle Mails die beantwortet wurden, werden, falls vorhanden, in den Ordner errorMessages verschoben. Die Mail, die den Prozess weitergeschickt hat wird in den konfigurierten Ordner verschoben.

Derjenige, der auf die Email antwortet, muss entweder direkter Bearbeiter der Aufgabe sein oder sich in der aktuell eingetragenen Gruppe befinden.

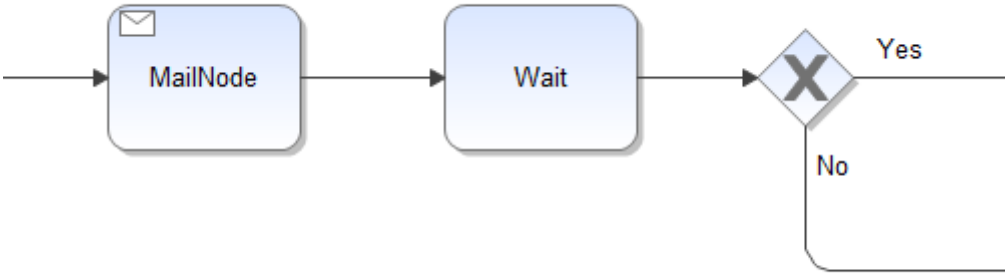
Beispiele



Parameter:

```
johnm1477@gmail.com,123456,993,imap.gmail.com,,SignalByMailReply mk4,-+{(Wait.*)}-+,true,decision,no:abgelehnt-nein-nok;jo:genehmigt-ok-ja,(.*?)schrieb ?s
```

Beispielprozess:



Einfaches Beispiel für den Mailtext:

```
Hallo,

für Entscheidung "Nein" antworten Sie bitte mit "abgelehnt", "nein", oder
"nok".

Für Entscheidung "Ja" antworten Sie bitte mit "genehmigt", "ja", oder "ok".

Viele Grüße
Ihr TIM-System

-----(${SYS.PROCESSINSTANCE_ID})-----{Wait4}-----
```

Abhängigkeiten

TIM Version : in dieser Konfiguration ab Version 3.6.2