Event - Job

Allgemein

Regeln des Typs "Job" können nur Fristen zugewiesen werden und müssen das Interface "JobRule" implementieren.

Innerhalb des Baums mit der Regelbibliothek werden sie dem Knoten "Job" zugeschrieben.

Struktur

Quellcode einer Klasse des Typs "Job":

```
package org.nuclet.lager;

import org.nuclos.api.rule.JobRule;
import org.nuclos.api.context.JobContext;
import org.nuclos.api.annotation.Rule;
import org.nuclos.api.exception.BusinessException;

/** @name
  * @description
  * @usage
  * @change
*/

@Rule(name="Job Lager", description="Job Lager")
public class JobLager implements JobRule {

 public void execute(JobContext context) {

//So werden Logs während des Ausführen in den Reiter Log Info in der Jobsteuerung geschrieben
                        context.joblog(e.getMessage());
            context.joblogError("TEST");
            context.joblogWarn("Warnung");
    }
}
```

Das entsprechende Interface schreibt die Implementierung der Methode "execute" vor und liefert als Kontext die Klasse JobContext. Darin enthalten ist die Session-Id.

Innerhalb der *execute*-Methode kann der Regelprogrammierer auch eine BusinessException auslösen, die zwar geloggt wird, aber nicht zum Abbruch der Jobdurchführung führt.

Die Angabe der Annotation "Rule" ist nicht zwingend notwendig, wird aber empfohlen, da die Beschreibungen (name und description) im Regelbibliotheksbaum angezeigt werden.

Transaktionale Jobs

Normalerweise wird die gesamte Verarbeitung des Jobs in nur einer Transaktion ausgeführt. Wenn man die Verarbeitung auf mehrere Transaktionen verteilen möchte muss das Interface TransactionalJobRule zusätzlich implementiert werden. Die Methode getTransactionalObjects(JobContext context) liefert eine Liste vom Typ Object wobei nun für jedes Element in der Liste eine eigene Transaktion eröffnet wird. Dabei wird für jedes Object die Methode execute(JobContext context) aufgerufen. Das Object kann vom JobContext mittels .getTransactionalObject() abgeholt werden.

Struktur TransactionalJobRule

Quellcode einer Klasse des Typs "Job" mit "TransactionalJobRule":

```
package example.rest;

import java.util.ArrayList;
import java.util.List;

import org.nuclos.api.annotation.Rule;
import org.nuclos.api.context.JobContext;
import org.nuclos.api.exception.BusinessException;
import org.nuclos.api.provider.QueryProvider;
import org.nuclos.api.rule.JobRule;
import org.nuclos.api.rule.TransactionalJobRule;

/** @name
  * @description
  * @usage
  * @change
*/

@Rule(name="OrderJob", description="Example transactional job rule")
public class OrderJob implements JobRule, TransactionalJobRule {

        public void execute(JobContext context) {
                // encapsulated transaction begin
                Order myOrder = (Order) context.getTransactionalObject();
                // do something ...

                // encapsulated transaction end
        }

        public List<Object> getTransactionalObjects(JobContext context) {
                List<Object> result = new ArrayList<Object>();
                try {
                        result.addAll(QueryProvider.getByState(Order.class, ExampleorderSM.State_10));
                } catch (BusinessException e) {
                        context.joblogError(e.getMessage());
                }
                return result;
        }
}
```

Mandantenabhängige Verarbeitung in Jobs

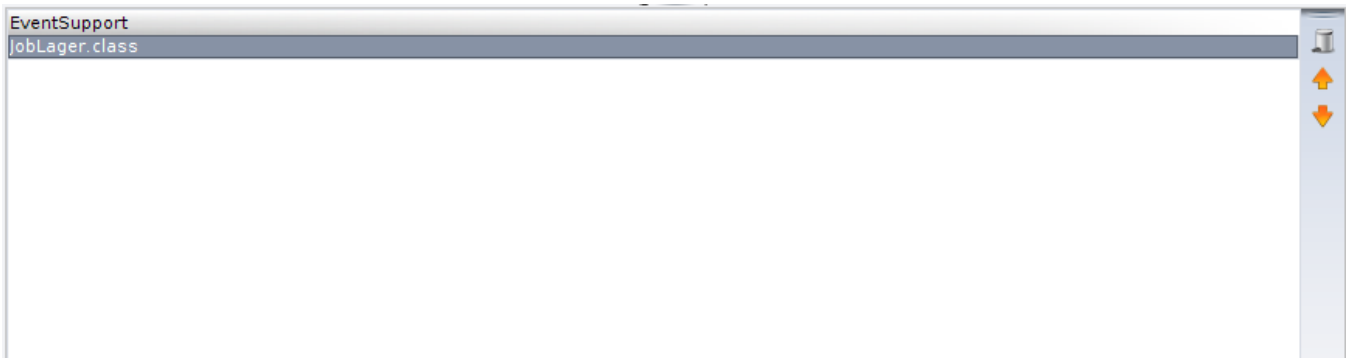
Wird die Mandantenfähigkeit für ein BO aktiviert, werden automatisch alle objektbezogenen Regeln (Speichern, Status, etc.) im Kontext des im BO hinterlegten Mandanten ausgeführt. Damit werden z.B. Queries auf andere BOs, welche ebenfalls mandantenabhängig sind, automatisch auf den gleichen Mandanten eingeschränkt.

Da ein Job nicht objektbezogen arbeitet kann in der Jobsteuerung gepflegt werden, für welche Mandanten dieser Job laufen soll. Ein Joblauf wird dann alle eingetragenen Mandanten berücksichtigen und hintereinander die execute Methode mit einem entsprechenden mandanteneingeschränkten Kontext aufrufen.

Eine weitere Alternative bietet die TransactionalJobRule (siehe weiter oben), wenn die Liste mandantenabhängige BOs enthält. In solch einem Fall ist die neue Transaktion auch gleich auf den im BO hinterlegten Mandanten eingeschränkt. Das hat den Vorteil, dass keine zusätzliche Konfiguration im Job nötig ist.

Zuweisung

Klassen des Typs "Job" können nur Fristen zugewiesen werden. Nach der Verknüpfung erscheint die Klasse nach Auswahl der Frist und des Typs im Zuweisungsfenster.



Mit dem Symbol des Mülleimers kann die Zuweisung der Klasse zur Frist gelöscht werden. Wichtig ist, dass damit nur die Zuweisung gelöscht wird. Weiterhin lässt sich mit Hilfe der Pfeile-Buttons die Ausführungsreihenfolge verändern.

Jede Änderung wird automatisch gespeichert.

Beispiel

Hier finden Sie ein Beispiel, das den Aufbau einer Regel vom Typ "Job" veranschaulicht.

Probleme und Lösungen

Support und Informationen zur Problembehandlung finden Sie hier.