

Event - Job

🌐 Sprache: **Deutsch** · [English](#)

📖 Event - Job

Ereignis-Regel „Job“: Interface, Kontext und ein Beispiel – so reagierst du auf dieses Nuclos-Ereignis.

REFERENZ

ENTWICKLER

STAND: JUL 2026

GILT FUER NUCLOS 4.2026.X

Auf dieser Seite

- [Wann greift die Regel?](#)
- [Struktur](#)
- [Zuweisung](#)
- [Beispiel](#)
- [Verwandte Seiten](#)

Wann greift die Regel?

Diese [serverseitige Regel](#) wird ausgeführt zeitgesteuert durch einen **Job/Timer**. Sie kann einem **Timer/Job** zugewiesen werden.

Struktur

Grundgerüst der Regelklasse (der Regeleditor legt es automatisch an):

```
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");
    }
}
```



Kontext: JobContext

Der Kontext JobContext liefert das betroffene BusinessObject über `context.getBusinessObject(...)`; zusätzlich stehen die Funktionen des RuleContext (Provider) bereit. Eine BusinessException bricht den Vorgang ab.

Zuweisung

Im [Server-Regelmanager](#) per Drag-and-Drop dem Ziel zuweisen. Optional lässt sich die Ausführung auf einen **Status** oder eine **Aktion** einschränken; die Reihenfolge steuerst du über die Pfeil-Buttons.



Beispiel

Das folgende Beispiel soll den Aufbau einer Regel vom Typ „Job„ veranschaulichen.

```
package org.nuclet.company;

import java.util.List;

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

/** @name
 * @description
 * @usage
 * @change
 */
@Rule(name="JobBestellungenPruefen", description="JobBestellungenPruefen")
public class JobBestellungenPruefen implements JobRule {

    public void execute(JobContext context){
        // Get all Bestellungen
        List<Bestellung> lstBestellungen = QueryProvider.execute(
            QueryProvider.create(Bestellung.class));

        for (Bestellung b : lstBestellungen) {
            // orders that are not confirmed are ok, but should be logged as infos
            if (Boolean.FALSE.equals(b.getBestaetigt())) {
                context.joblog("Info: Bestellung " + b.getNummer() + " wurde noch nicht bestätigt.");
            }

            // orders without articles are formally valid, but dont make sense
            if (b.getPosition().size() <= 0) {
                context.joblogWarn("Warnung: Bestellung " + b.getNummer() + " besitzt keine Positionen");
            }

            // orders without an internal advisor are formally invalid
            if (b.getSachbearbeiter() == null) {
                context.joblogError("Fehler: Bestellung " + b.getNummer() + " besitzt keinen Sachbearbeiter");
            }
        }
    }
}
```

Verwandte Seiten

Regeln (serverseitig)

Grundlagen.

[Öffnen](#)

Server Regelmanager

Zuweisen.

[Öffnen](#)

Ausführungsreihenfolge der Regeln

Reihenfolge.

[Öffnen](#)