

Event - Statuswechsel (im Anschluss)

🌐 Sprache: **Deutsch** · [English](#)

📖 Event - Statuswechsel (im Anschluss)

Ereignis-Regel „Statuswechsel (im Anschluss)“: 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 **nach** einem erfolgreichen Statuswechsel. Sie kann einem **Statusmodell** zugewiesen werden.

Struktur

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

```
package org.nuclet.lager;

import org.nuclos.api.rule.StateChangeFinalRule;
import org.nuclos.api.context.StateChangeContext;
import org.nuclos.api.annotation.Rule;
import org.nuclos.api.exception.BusinessException;

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

@Rule(name="Statuswechsel im Anschluss Lagerposition", description="Statuswechsel im Anschluss Lagerposition")
public class StatuswechselImAnschlussLagerposition implements StateChangeFinalRule {

    public void changeStateFinal(StateChangeContext context) throws BusinessException {
    }
}
```



Kontext: StateChangeContext

Der Kontext StateChangeContext 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 „Statuswechsel (im Anschluss) und die Weiterverarbeitung der übergebenen Inhalte veranschaulichen.

```
package org.nuclet.company;

import org.nuclos.api.annotation.Rule;
import org.nuclos.api.context.StateChangeContext;
import org.nuclos.api.exception.BusinessException;
import org.nuclos.api.provider.StatemodelProvider;
import org.nuclos.api.rule.StateChangeFinalRule;

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

@Rule(name="Auftrag beantragen", description="Prüfung und Weiterführung eines neuen Antrags")
public class Auftragbeantragen implements StateChangeFinalRule {
    public void changeStateFinal(StateChangeContext context) throws BusinessException {

        Auftrag neuerAuftrag = context.getBusinessObject(Auftrag.class);

        // Wurde noch keine Auftragsnummer hinterlegt, kann der Auftrag nicht
        // ausgeführt werden. Wir brauchen eine weitere Prüfung
        if (neuerAuftrag.getAuftragsnr() == null) {
            // Neuer Status: Prüfen (Numeral 55)
            StatemodelProvider.changeState(neuerAuftrag, ProzessAuftragSM.State_55);
        }
        else {
            // Prüfung ok, neuer Status: beauftragt (Numeral 30)
            StatemodelProvider.changeState(neuerAuftrag, ProzessAuftragSM.State_30);
        }
    }
}
```

Verwandte Seiten

Regeln (serverseitig)

Grundlagen.

[Öffnen](#)

Server Regelmanager

Zuweisen.

[Öffnen](#)

Ausführungsreihenfolge der Regeln

Reihenfolge.

[Öffnen](#)