

Event - Aktualisieren (im Anschluss)

 Sprache: **Deutsch** · [English](#)

Event - Aktualisieren (im Anschluss)

Ereignis-Regel „Aktualisieren (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** dem erfolgreichen Aktualisieren eines Datensatzes. Sie kann einem **Businessobjekt** zugewiesen werden.

Struktur

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

```
package org.nuclet.lager;

import org.nuclos.api.rule.UpdateFinalRule;
import org.nuclos.api.context.UpdateContext;
import org.nuclos.api.annotation.Rule;
import org.nuclos.api.exception.BusinessException;

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

@Rule(name="Aktualisieren Lagerposition im Anschluss", description="Aktualisieren Lagerposition im Anschluss")
public class AktualisierenLagerPositionImAnschluss implements UpdateFinalRule {

    public void updateFinal(UpdateContext context) throws BusinessException {
    }
}
```



Kontext: UpdateContext

Der Kontext UpdateContext 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.

Regel	Status	Aktion
AktualisierenLagerPositionImAnschl...		




Zuweisung „Aktualisieren (im Anschluss)“.

Beispiel

Das folgende Beispiel soll den Aufbau einer Regel vom Typ „Aktualisieren (im Anschluss)“, und die Weiterverarbeitung der übergebenen Inhalte veranschaulichen.

```
package org.nuclet.company;

import java.util.List;
import org.nuclos.api.rule.UpdateRule;
import org.nuclos.api.context.UpdateContext;
import org.nuclos.api.annotation.Rule;
import org.nuclos.api.exception.BusinessException;

/** @name
 * @description
 * @usage
 * @change
 */
@Rule(name="UpdateFinalBestellung", description="UpdateFinalBestellung")
public class UpdateFinalBestellung implements UpdateFinalRule {
    public void updateFinal(UpdateContext context) throws BusinessException {

        // this is the businessObject after storing the updated data in the database
        Bestellung curBestellung = context.getBusinessObject(Bestellung.class);

        // changes are not recognized, because data has already been saved
        // the following description will not be stored in the database
        curBestellung.setKurzbeschreibung("Bestellung wurde im Schnellverfahren abgeschlossen.");

        // we want to re-update all positions after the order has been updated properly
        // so we use the QueryProvider
        for (Bestellposition pos : curBestellung.getPosition()) {
            pos.setBemerkung("Position im Gesamtkontext von " + curBestellung.getNr());
            pos.save();
        }
    }
}
```

Verwandte Seiten



Regeln (serverseitig)

Grundlagen.

[Öffnen](#)



Server Regelmanager

Zuweisen.

[Öffnen](#)



Ausführungsreihenfolge der Regeln

Reihenfolge.

[Öffnen](#)

