

Regeln (serverseitig)

 Sprache: **Deutsch** · [English](#)

Regeln (serverseitig)

Java-Geschäftsregeln auf dem Server: an Ereignisse geknüpft, mit Zugriff auf Businessobjekte, Provider und die volle Nuclos-API.

REFERENZ

ENTWICKLER

STAND: JUL 2026

GILT FUER NUCLOS 4.2026.X

Auf dieser Seite

- [Was sind serverseitige Regeln?](#)
- [Ereignisse](#)
- [Regeleditor](#)
- [Anwendungsspezifische Objekte](#)
- [Provider \(unterstützende Klassen\)](#)
- [Regeln, Unterformulare und der Regelkontext](#)
- [Verwandte Seiten](#)

Was sind serverseitige Regeln?

Die Nuclos-API ermöglicht Java-basierte Geschäftsregeln, die bei bestimmten **Ereignissen** lesend und schreibend auf Inhalte zugreifen. Reichen die übrigen Low-Code-Mittel nicht aus, setzt du hier komplexe Validierungen, Berechnungen, Webservice-Aufrufe oder automatische Datenänderungen um.

 **API**

Die vollständige API ist unter api.nuclos.de dokumentiert.

Menüaufruf: *Konfiguration* *Regelwerke* *Regeln (serverseitig)*

Ereignisse

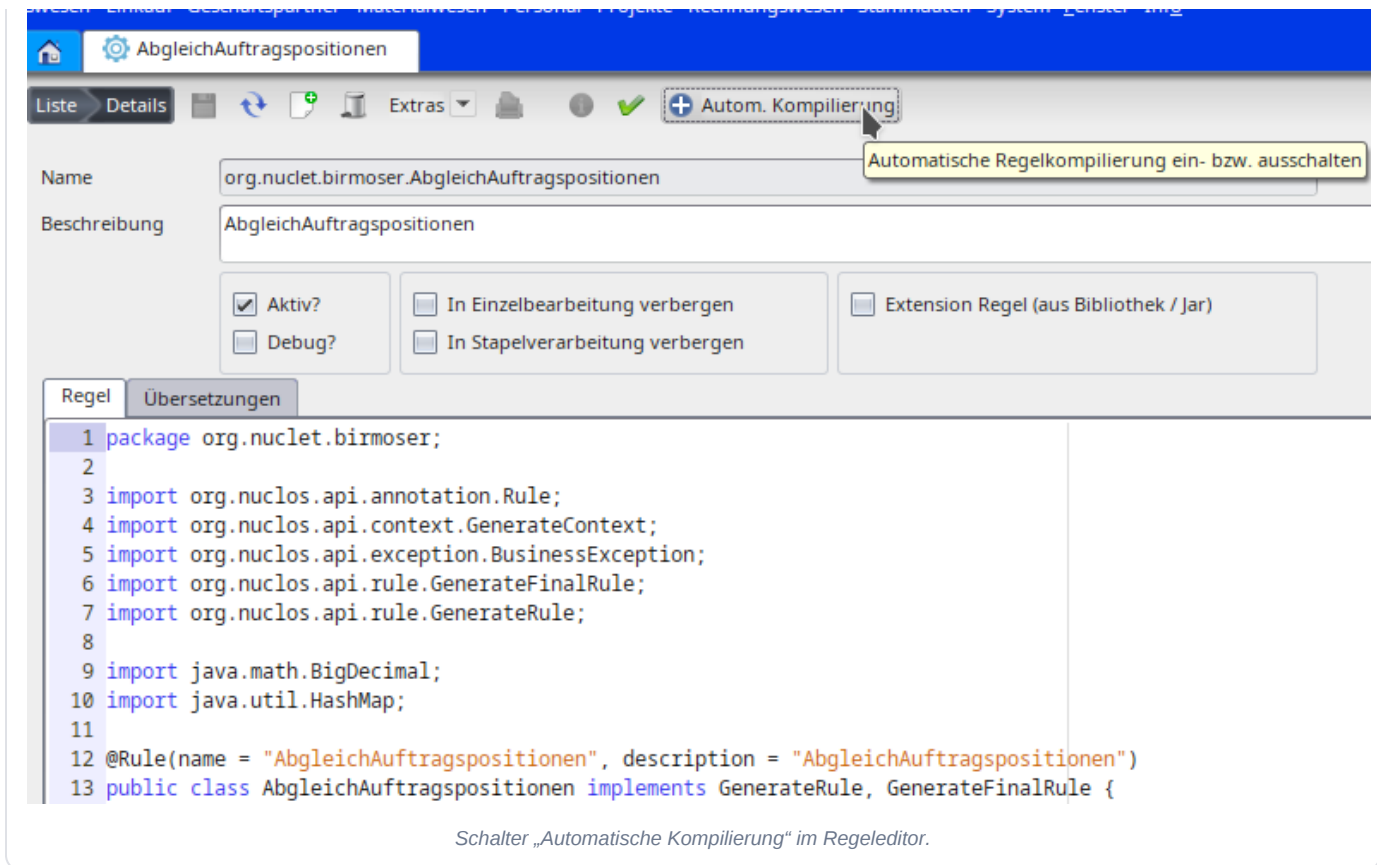
Ereignis	Regel-Interface(s)	Beispiel
Neuanlage	InsertRule, InsertFinalRule	neuer Datensatz (z. B. Rechnung)
Änderung	UpdateRule, UpdateFinalRule	bestehender Datensatz (z. B. Auftrag)
Löschen	DeleteRule, DeleteFinalRule	bestehender Datensatz
Statuswechsel	StateChangeRule, StateChangeFinalRule	z. B. Offen Mahnung
Objektgenerator	GenerateRule, GenerateFinalRule	z. B. Lieferschein aus Auftrag
Schaltfläche	CustomRule	Benutzeraktion im Layout
Job	JobRule	zeitgesteuert, wiederkehrend

Regeleditor

Der Regeleditor ist das zentrale Werkzeug zum Bearbeiten von Regeln. Für komfortable Entwicklung/Debugging gibt es zusätzlich die IntelliJ-Anbindung.

 **Automatische Kompilierung**

Über den Schalter **Automatische Kompilierung** lässt sich das Kompilieren beim Speichern vorübergehend abschalten (nützlich bei voneinander abhängigen, noch nicht kompilierbaren Regeln). **Achtung:** Nicht kompilierte Regeln sind im [Regelmanager](#) nicht sichtbar.



AbgleichAuftragspositionen

Automatische Regelkompilierung ein- bzw. ausschalten

Name: org.nuclet.birmoser.AbgleichAuftragspositionen

Beschreibung: AbgleichAuftragspositionen

☒ Aktiv? ☐ In Einzelbearbeitung verbergen ☐ Extension Regel (aus Bibliothek / Jar)

☐ Debug? ☐ In Stapelverarbeitung verbergen

Regel Übersetzungen

```

1 package org.nuclet.birmoser;
2
3 import org.nuclos.api.annotation.Rule;
4 import org.nuclos.api.context.GenerateContext;
5 import org.nuclos.api.exception.BusinessException;
6 import org.nuclos.api.rule.GenerateFinalRule;
7 import org.nuclos.api.rule.GenerateRule;
8
9 import java.math.BigDecimal;
10 import java.util.HashMap;
11
12 @Rule(name = "AbgleichAuftragspositionen", description = "AbgleichAuftragspositionen")
13 public class AbgleichAuftragspositionen implements GenerateRule, GenerateFinalRule {

```

Schalter „Automatische Kompilierung“ im Regeleditor.

Anwendungsspezifische Objekte

Nuclos generiert automatisch repräsentierende Objekte für Businessobjekte, Statusmodelle, Datenquellen und Reports (mit Gettern/Settern). Ändern sich diese Metadaten, lassen sich abhängige Regeln nicht mehr kompilieren – ein direkter Hinweis auf anzupassende Stellen.



BigDecimal

Kommazahl-Attribute werden als `java.math.BigDecimal` abgebildet. Nutze `BigDecimal.valueOf(double)` statt `new BigDecimal(double)` und bei Division stets `.divide(BigDecimal, RoundingMode)`.

Provider (unterstützende Klassen)

Provider bieten Funktionen, um Inhalte aus Regeln heraus zu verändern: **QueryProvider**, **StatemodelProvider**, **GenerationProvider**, **DatasourceProvider**, **BusinessObjectProvider**, **MailProvider**, **PrintoutProvider**, **ReportProvider**, **FileProvider**.

Regeln, Unterformulare und der Regelkontext

- In Regeln **vor** dem Ereignis (Insert/Update/StateChange/Generate) werden Änderungen am Kontextobjekt über Attribut-Setter automatisch übernommen; `delete()` auf abhängige Objekte gilt als Änderung.
- In **Final**-Regeln (im Anschluss) ist zum Ändern anderer Objekte ein explizites `save()` nötig – das löst deren Update-/UpdateFinal-Regeln aus.



Endlosschleifen

Ein `save()` auf das Kontextobjekt in einer UpdateFinalRule kann eine **Endlosschleife** auslösen – ggf. [Regelausführung selektiv ausschalten](#) oder [SaveFlags](#) verwenden.

Verwandte Seiten



Regelwerke

Übersicht.



Server Regelmanager

Regeln zuweisen.



Ausführungsreihenfolge der Regeln

Öffnen

Öffnen

Reihenfolge je Aktion.

Öffnen