Regeln (serverseitig)

Definition

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

Die Nuclos API ermöglicht die Erstellung von (java-basierenden) Geschäftsregeln, die lesend und schreibend bei bestimmten Ereignissen auf Inhalte in Nuclos zugreifen können. Dazu bietet die Nuclos API verschiedene Klassen und Objekte an, um mit Nuclos zu interagieren.

> Die vollständige Dokumentation der Nuclos API findet sich unter http://api.nuclos.de/

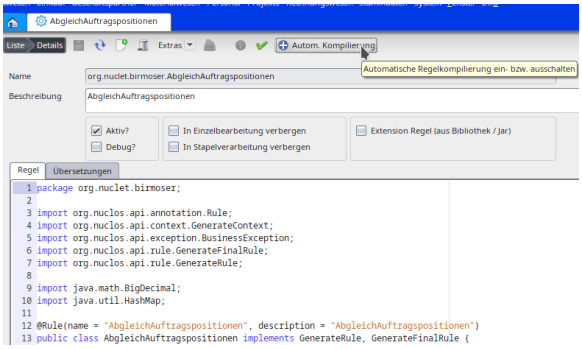
Ereignisse

Ereignisse (Ereignisse Event Regeln), an die Geschäftsregeln geknüpft werden können, sind:

- Die **Neuanlage** (InsertRule, InsertFinalRule) eines neuen Datensatzes (z.B. einer Rechnung, etc.)
- Die **Änderung** (UpdateRule, UpdateFinalRule) eines existierenden Datensatzes (z.B. eines Auftrages, etc.)
- Das **Löschen** (DeleteRule, DeleteFinalRule) eines existierenden Datensatzes (z.B. eines nicht mehr benötigten Adressdatensatzes, etc.)
- Der **Statuswechsel** (StateChangeRule, StateChangeFinalRule) in einem existierenden Datensatz (z.B. die Überführung einer Rechnung von "Offen" nach "Mahnung", etc.)
- Die Durchführung eines **Objektgenerators** (GenerateRule, GenerateFinalRule) (z.B. die Erstellung eines Lieferscheines aus einem Auftrag, etc.)
- Das Betätigen einer **Schaltfläche** (CustomRule) durch den User (z.B. um bestimmte Aktionen auszulösen, etc.)
- Automatische **Jobs** (JobRule), die regelmässig ausgeführt werden können (z.B. die nächtliche Aktualisierung eines Onlineshops, etc.)

Regeleditor

Der Regeleditor ist das zentrale Werkzeug in Nuclos, um Regeln zu bearbeiten. Aktuell gibt es ergänzend ein Tutorial: Einbinden von Nuclos in IntelliJ IDEA für die erleichterte Entwicklung und das erleichterte Debugging von Regel in Entwicklung.



> Um voneinander abhängige Regeln, die nicht kompilierbar sind, korrigieren zu können, steht der Schalter "Automatische Kompilierung" (in der Toolbar des Regeleditors) zur Verfügung. Darüber lässt sich die automatische Kompilierung beim Speichern vorübergehend deaktivieren. Sobald man sie dort wieder aktiviert, wird beim nächsten Speichern wieder alles durchkompiliert.
>
> Achtung: Solange Regeln nicht kompiliert sind, sind sie im Regelmanager (siehe folgender Abschnitt) auch nicht sichtbar.

Regelmanager

Der Server Regelmanager (*Menüaufruf: (Konfiguration) - (Server Regelmanager)*) ist das zentrale Werkzeug in Nuclos, um die Zuordnung zwischen Geschäftsregeln und Ereignissen zu verwalten. Die Zuweisung von Geschäftsregeln aus der Regelbibliothek (linker Baum) zu den Ereignissen (rechter Baum) geschieht mittels Drag&Drop.

Anwendungsspezifische Objekte

Nuclos kümmert sich automatisch um die Generierung aller nötigen anwendungsspezifischen Objekte. Dazu zählen

- **Businessobjekte** (https://api.nuclos.de/org/nuclos/api/businessobject/). Businessobjekte repräsentieren die konfigurierten Businessobjekt jedes Nuclets. Für jedes Businessobjekt wird ein repräsentierendes Businessobjekt erzeugt, das entsprechende getter- und setter-Methoden für sämtliche Attribute des Businessobjekts bereitstellt.
- **Statusmodelle** (https://api.nuclos.de/org/nuclos/api/statemodel/). Statusmodelle repräsentieren die konfigurierten Prozesse jedes Nuclets. Für jeden Prozess (Statusmodell) wird ein repräsentierendes Objekt erzeugt, dass den Geschäftsregeln einen erleichterten Zugriff auf die Statusnamen und -numerale bietet.
- **Datenquellen** (https://api.nuclos.de/org/nuclos/api/datasource/). Datenquellen repräsentieren die konfigurierten Datenquellen jedes Nuclets. Für jede Datenquelle wird ein repräsentierendes Objekt erzeugt, dass den Geschäftsregeln einen erleichterten Zugriff auf die Datenquelleninhalte bietet.
- **Reports** (https://api.nuclos.de/org/nuclos/api/report/)
- usw..

Werden Businessobjekte, Statusmodelle, Datenquellen, Reports und Formulare geändert, sind die Regeln, die diese anwendungsspezifischen Objekte verwenden, nicht mehr kompilierbar. Das hilft bei der unmittelbaren Erkennung der in Geschäftsregeln ggf. anzupassenden Stellen, die vorübergehend nicht mehr zu den geänderten anwendungsspezifischen Objekten passen.

> Attribute vom Datentyp "Kommazahl" werden in Businessobjekten grundsätzlich über java.math.BigDecimal abgebildet. Mehr Informationen zu BigDecimal siehe http://docs.oracle.com/javase/7/docs/api/java/math/BigDecimal.html.
>
> Zwei zeitsparende Hinweise zu BigDecimal:
>
> - Statt BigDecimal(double) immer BigDecimal.valueOf(Double) oder BigDecimal(String) verwenden.
> - Bei Divisionen immer .divide(BigDecimal, RoundingMode) statt .divide(BigDecimal) verwenden, um eine ArithmeticException zu vermeiden.

Unterstützende Klassen (Provider)

Provider (https://api.nuclos.de/org/nuclos/api/provider/) bieten unterstützende Funktionen, um Inhalte in Nuclos über Geschäftsregeln zu verändern. Aktuell sind folgende Provider implementiert.

- QueryProvider
- StatemodelProvider
- GenerationProvider
- DatasourceProvider
- BusinessObjectProvider
- MailProvider
- PrintoutProvider
- ReportProvider
- FileProvider

Regelausführung und Unterformulare

Die Regelausführung bei Änderung von BOs in Unterformularen bedarf derzeit noch der Systematisierung in Nuclos.

Derzeitigs Verhalten ist wie folgt:

- Bei **Neuanlage** (InsertRule, InsertFinalRule) **nur bei Objekten mit Statusmodell**
- Bei **Änderung** (UpdateRule, UpdateFinalRule) **keine** Regelausführung
- Beim **Löschen** (DeleteRule, DeleteFinalRule) **nur bei Objekten mit Statusmodell**

Ändern, Hinzufügen und Löschen von Businessobjekten im Regelkontext

- In Regeln die **vor** dem auslösenden Ereignis stattfinden (Generation-, Update-, Insert-, und Statechange-Rule) werden Änderungen am Kontextobjekt über Attribut-Setter automatisch übernommen

- das **Löschen von abhängigen Objekten** aus einem Unterformular über die Methode *delete()* gilt dabei als Änderung am Kontextobjekt
- In Regeln die **nach** dem auslösenden Ereignis stattfinden (GenerationFinal-, UpdatFinale-, InsertFinal- und StateChangeFinal-Rule) gilt das nicht
- In diesem Fall (und wenn abhängige Objekten aus einem Unterformular oder beliebige andere Objekte geändert werden sollen) ist ein Aufufen Metthode *save()* erforderlich
- Die Methode ***save()*** löst für das gespeicherte Objekt die zugehörigen Update- und UpdateFinaRules aus.
    - Achtung: Ein *save()* auf das Kontextobjekt in einer UpdateFinalRule kann eine Endlosschleife auslösen, wenn es keine andere Mögllichkeit gibt empfiehlt es sich die Regelausführung selektiv auszuschalten
- Das Einfügen von neu angelegten Businessobjekten über *insert ()* ist in jedem Regelkontext möglich