

Geschäftsregeln

Regeleditor

Menüaufruf: (Konfiguration) - (Regelwerke) - (Geschäftsregeln)

Regeln bearbeiten

[blocked URL](#)

In dieser Maske können Geschäftsregeln definiert werden. Vergeben Sie hier einen **Namen**, unter dem später die Regel auswählbar ist sowie eine **Beschreibung** nach Bedarf.

Ist das **Aktiv**-Flag gesetzt, wird die Regel über compiliert und die Binärdateien auf dem Server abgelegt.



Achtung: Nur eine aktive Regel wird auch ausgeführt !

Den Java-Sourcecode erfassen Sie im Reiter **Regel**. Wenn Sie die Regel speichern, wird ein Java Compiler angestossen. Im Falle eines Fehlers sehen Sie im unteren Bereich des Fensters die Compiler Ausgaben.

[blocked URL](#)

Regelausführung

Im Reiter **Verwendung** können Sie die Ausführung der Regel spezifizieren. Wählen Sie zunächst das **Businessobjekt**, für die die Regel ausgeführt werden soll. Unter **Ereignis** haben Sie folgende Auswahlmöglichkeiten:

- **Speichern:** Regel wird VOR dem Speichern eines Objektes ausgeführt
- **Löschen:** Regel wird VOR dem Löschen eines Objektes ausgeführt
- **Benutzer:** Regel kann manuell ausgeführt werden

[blocked URL](#)

Wenn Sie die 'Benutzer' eingestellt haben, können Sie die Regel direkt aus dem Datensatz heraus ausführen. Dazu betätigen Sie den Button **Regel ausführen**.

[blocked URL](#)

Wählen Sie in dem Dialog die entsprechende Regel aus. Möchten Sie das Objekt nach der Regelausführung gespeichert haben, so aktivieren Sie **Objekt nach Regelausführung speichern**.

Beispiele

Verfügbare Methoden zum Auslesen und Schreiben von Werten und ID-Feldern

```

// Wert auslesen bei Businessobjekt ohne Statusmodell:
Object wert = (Object)server.getFieldValue(sEntityName, sFieldName); // Wert aus aktuellem Objekt auslesen
Object wert = (Object)server.getFieldValue(sEntityName, iObjectId, sFieldName); // Wert aus Objekt iObjectId
auslesen
// Wert auslesen bei Businessobjekt mit Statusmodell:
Object wert = (Object)server.getAttribute(sAttributeName).getValue(); // Wert aus aktuellem Objekt auslesen
Object wert = (Object)server.getAttribute(iObjectId, sAttributeName).getValue(); // Wert aus Objekt iObjectId
auslesen

// Wert setzen bei Businessobjekten mit und ohne Statusmodell
server.setFieldValue(sEntityName, sFieldName, oValue); // Wert in aktuellem objekt setzen
server.setFieldValue(sEntityName, iObjectId, sFieldName, oValue); // Wert in Objekt iObjectId setzen

// ID auslesen bei Businessobjekt mit Statusmodell:
Integer id = server.getAttribute(sAttributeName).getValueId(); // Id-Feld in aktuellem objekt auslesen
Integer id = server.getAttribute(iGenericObjectId, sAttributeName).getValueId(); // Id-Feld aus Objekt
iObjectId auslesen

// ID setzen bei Businessobjekt mit und ohne Statusmodell:
server.setField(sEntityName, sFieldName, iValueId, oValue); // Id-Feld in aktuellem objekt setzen
server.setField(sEntityName, iObjectId, sFieldName, iValueId, oValue); // Id-Feld in Objekt iObjectId setzen

// Untergeordnetes Businessobjekt -> Unterformular bei Businessobjekt mit Statusmodell:
Collection<MasterDataVO> collmdvo = server.getDependants(sEntityName);

// Untergeordnetes Businessobjekt -> Unterformular bei Businessobjekt ohne Statusmodell:
Collection<MasterDataVO> collmdvo = server.getDependants(sEntityName, null);

Collection<MasterDataVO> collmdvo = server.getDependants(iObjectId, sEntityName, sForeignKeyFieldName); // UF
in Objekt iObjectId, übergeben Sie hier zusätzlich den Namen des Referenzfeldes auf die übergeordnetes
Businessobjekt (sForeignKeyFieldName)

for (MasterDataVO mdvo : collmdvo) {
    // Wert auslesen
    Object wert = (Object)mdvo.getField(sFieldName); // Beispiel: Übergabeparameter = 'kunde'
    // ID auslesen
    Integer id = (Integer)mdvo.getField(sFieldName+"Id"); // Beispiel: Übergabeparameter = 'kundeId'

    // Wert setzen
    mdvo.setField(sFieldName, oValue);
    // ID setzen
    mdvo.setField(sFieldName+"Id", oValue);
}

```



Achtung Sonderfall:

Soll eine ID in einem MasterData Objekt eingetragen werden, dass nicht über den Hauptdatensatz (mit getDependants) geholt wurde, sondern z.B. so:

```

MasterDataVO item = server.getMasterData("<EntityName>", iObjectId);
item.setField("<fieldName>", oValue);
item.setField("<fieldName>" + "Id", iValueId);

```

so werden Änderungen nicht übernommen, weil nur im Object Item gespeichert wird und nicht in der Datenbank. Um dies zu erreichen muss das Objekt zum Speichern an Nuclos explizit übergeben werden:

```

        MasterDataVO item = server.getMasterData("<EntityName>", iObjectId);
item.setField("<fieldName>" + Id , iValueId);
    try {
        final org.nuclos.server.masterdata.ejb3.MasterDataFacadeLocal mdFa cade = ServiceLocator.getInstance() .
getFacade(org.nuclos.server.masterdata.ejb3.MasterDataFacadeLocal.class);
        mdFacade.modify("<EntityName>", item , null);
    }
    catch(Exception e) {
        throw new NuclosBusinessRuleException(e);
    }
}

```

Neue Zeilen in Unterformular hinzufügen

```

        Map<String, Object> mpFields = new HashMap();
mpFields.put("<Feld1>", <Wert>);
mpFields.put("<Feld2>", <Wert>);
...
server.addSubformEntry("<Name Subbusinessobjekt>", mpFields);

```

MasterDataVO Objekte löschen

```

        MasterDataVO mdvo = ....;
try {
org.nuclos.common2.ServiceLocator.getInstance().getFacade(org.nuclos.server.masterdata.ejb3.
MasterDataFacadeLocal.class).remove("<Name des Businessobjekts>", mdvo, false);
}catch (Exception e) {
    e.printStackTrace();
}

```

Dokumentenanhänge auslesen und bearbeiten

```

        public void rule(RuleInterface server) throws NuclosBusinessRuleException {
final Integer intid = server.getGenericObject().getId();
// Bedingung
final CollectableComparison condDocs =
        org.nuclos.common.SearchConditionUtils.newMDReferenceComparison(MasterDataMetaCache.
getInstance().getMetaData(org.nuclos.common.NuclosEntity.GENERALSEARCHDOCUMENT.getEntityName()),
        "genericObject", intid);

// Suchbedingung zusammenbauen (CollectableSearchExpression)
org.nuclos.server.genericobject.searchcondition.CollectableSearchExpression expression = new org.nuclos.
server.genericobject.searchcondition.CollectableSearchExpression(condDocs);
Collection<Object> result = server.getMasterDataIds(org.nuclos.common.NuclosEntity.GENERALSEARCHDOCUMENT.
getEntityName(), expression);
// Datensatz auslesen
final MasterDataVO mdvoDocument = server.getMasterData(org.nuclos.common.NuclosEntity.GENERALSEARCHDOCUMENT.
getEntityName(), org.nuclos.common2.IdUtils.unsafeToId(result.iterator().next()));
// Datei extrahieren
final org.nuclos.server.genericobject.valueobject.GenericObjectDocumentFile file = mdvoDocument.getField
("file", org.nuclos.server.genericobject.valueobject.GenericObjectDocumentFile.class);
}

```