

# Fristenregeln

Fristenregeln werden verwendet, um Aktionen auf bestimmten Mengen von Datensätzen durchzuführen. Die Ausführung der Fristenregel wird über die [Jobsteuerung](#) konfiguriert.

## Fristenregel anlegen

Der Inhalt einer Fristenregel unterscheidet sich von der Geschäftsregel durch die Methoden.

- `public Collection<Integer> getIntIds(RuleInterface server)`

In dieser Methode muss zunächst eine Menge von Datensätzen (Ids) gesammelt und als Collection zurück gegeben werden.

- `public void process(RuleInterface server, Integer ild)`

In dieser Methode werden die Daten dieser Collection nacheinander abgearbeitet. Hier findet also die eigentliche Aktion auf das Objekt statt.

[blocked URL](#)

## Beispiel

Hier werden zunächst alle Verträge im Status "Vertrag läuft" gesucht und in der Collection ids zusammengefasst. Danach wird nach Eintritt eines bestimmten Datums ein Statuswechsel im Vertrag durchgeführt. Das *server* Objekt ist dabei das gerade bearbeitete Objekt *ild*.

[blocked URL](#)

## RuleInterface: SearchConditions für Objektsuche

Um in der Methode `getIntIds` auf die benötigte Menge Datensätze zu kommen, können die folgenden Methoden verwendet werden. Die sogenannten SearchConditions arbeiten wie die manuelle Suche über die Suchmaske.

### Businessobjekte ohne Statusmodell

```
// Suchbedingungen für Feld- bzw. Wertevergleiche
CollectableComparison cond1 = org.nuclos.common.SearchConditionUtils.newMDComparison(org.nuclos.server.common.MasterDataMetaCache.getInstance().getMetaData("<Businessobjekt>"), "<Feld>", ComparisonOperator.EQUAL, "<Suchwert>");
CollectableComparison cond2 = org.nuclos.common.SearchConditionUtils.newMDComparison(org.nuclos.server.common.MasterDataMetaCache.getInstance().getMetaData("<Businessobjekt>"), "<Feld>", ComparisonOperator.LESS_OR_EQUAL, server.now());

// Suchbedingung für ID Felder
CollectableComparison cond3 = org.nuclos.common.SearchConditionUtils.newMDReferenceComparison(MasterDataMetaCache.getInstance().getMetaData("<Businessobjekt>"), "<ID-Feld>", <Id>);

CollectableComparison cond4 = ....
...

// Suche ausführen
Collection<Object> ids = server.getMasterDataIds("<Businessobjekt>", new org.nuclos.server.genericobject.searchcondition.CollectableSearchExpression(server.newAndCondition(cond1, cond2, cond3,...)));
```

### Businessobjekte mit Statusmodell

```
CollectableSearchCondition cond1 = server.newGOComparison("<Feld>", ComparisonOperator.EQUAL, <Suchwert>);
CollectableSearchCondition cond2 = ...

Collection<Integer> ids = server.getGenericObjectIds("<Businessobjekt>", cond1);
// ODER
Collection<Integer> ids = server.getGenericObjectIds("<Businessobjekt>", server.newAndCondition(cond1, cond2));
```

## Verfügbare Vergleichsoperatoren

```
ComparisonOperator.EQUAL
ComparisonOperator.LESS
ComparisonOperator.GREATER
ComparisonOperator.LESS_OR_EQUAL
ComparisonOperator.GREATER_OR_EQUAL
ComparisonOperator.NOT_EQUAL
ComparisonOperator.LIKE
ComparisonOperator.NOT_LIKE
ComparisonOperator.IS_NULL
ComparisonOperator.IS_NOT_NULL
```

## Verfügbare Suchvariationen

```
server.newAndCondition(CollectableSearchCondition... acond)
server.newOrCondition(CollectableSearchCondition... acond)
server.newNotCondition(CollectableSearchCondition... acond)
```

## Protokollierung

Die Methoden

```
logError(java.lang.String sMessage)
logInfo(java.lang.String sMessage)
logWarning(java.lang.String sMessage)
```

schreiben die entsprechenden Ausgaben in das Protokoll der [Jobsteuerung](#).