

# BusinessObjectProvider

🌐 Sprache: **Deutsch** · [English](#)

## BusinessObjectProvider

BOs aus Regeln anlegen, ändern und löschen – `save()`, \*All-Methoden und logisches Löschen.

REFERENZ

ENTWICKLER

STAND: JUL 2026

GILT FUER NUCLOS 4.2026.X

### Auf dieser Seite

- [Wozu dient der BusinessObjectProvider?](#)
- [save\(\) – anlegen & ändern](#)
- [insertAll / updateAll / deleteAll](#)
- [deleteLogical / deleteLogicalAll](#)
- [Verwandte Seiten](#)

## Wozu dient der BusinessObjectProvider?

Die Klasse `BusinessObjectProvider` stellt Funktionen zum **Anlegen, Ändern und Löschen** von Businessobjekten aus einer Regel heraus bereit – auch für Objekte, die nicht zum aktuellen Ablauf (Kontext) gehören.



### Achtung: parallele Bearbeitung

Bereits in Masken dargestellte Daten werden über den Kontext in die Regel geladen. Werden dieselben Daten parallel über den `BusinessObjectProvider (update)` verarbeitet, kann es durch unterschiedliche Bearbeitungsversionen zu Fehlern kommen.



### Moderner Weg

Die statischen Methoden `insert`, `update` und `delete` sind **deprecated**. Nutze stattdessen das Interface `Modifiable` (`save()`/`delete()` direkt am BO).

## save() – anlegen & ändern

Neue BOs mit `new` instanzieren, befüllen und mit `save()` speichern. Referenzen (z.B. Artikel, Lager) über ihre Id zuweisen – z.B. per [QueryProvider](#).

```
public class Bestellanlegen implements InsertRule {
    public void insert(InsertContext context) throws BusinessException {

        Anfrageposition newPos = new Anfrageposition();

        Artikel myArticel = QueryProvider.getById(Artikel.class, 40465351L);
        Lager myLager = QueryProvider.getById(Lager.class, 40276065L);

        newPos.setPositionsnr(0);
        newPos.setArtikelId(myArticel.getId());
        newPos.setAnzahl(2.0d);
        newPos.setLagerId(myLager.getId());

        newPos.save();
    }
}
```

## insertAll / updateAll / deleteAll

Verarbeiten eine `Collection` sequentiell und reichen jeden Eintrag an die jeweilige Einzelmethode weiter.

```
public static <T extends BusinessObject> void insertAll(Collection<T> type) throws BusinessException;
```

## deleteLogical / deleteLogicalAll

Löschen Einträge **logisch** (nur für BOs, die LogicalDeletable implementieren). Bestehende Verknüpfungen können ein Löschen verhindern und die Regelverarbeitung abbrechen.

QueryProviders

## Verwandte Seiten

### QueryProvider

Daten laden.

[Öffnen](#)

### DatasourceProvider

Datenquellen ausführen.

[Öffnen](#)

### GenerationProvider

Arbeitsschritte.

[Öffnen](#)