

BusinessObjectProvider (Beispiele)

Anlegen neuer Einträge

import org.nuclos.api.provider.BusinessObjectProvider;

Methode	Beispiel
save	<pre>public class Bestellungenanlegen 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(); } }</pre> <p>Erläuterung:</p> <p>Neue BusinessObjekte werden einfach mit <i>new</i> instantiiert und anschließend befüllt. Denken Sie daran, dass viele Instanzen von Businessobjekten in Nuclos Referenzen auf andere Businessobjekte besitzen. In oberen Beispiel das Lager und der Artikel. In diesem Fall ermitteln Sie die Daten, z.B. mit Hilfe des QueryProviders, und weisen sie über die Id des Eintrages zu. Wurde das BusinessObjekt korrekt und vollständig befüllt, kann es mit dem Aufruf der Methode "insert" abgespeichert werden.</p>
insertAll	<p>Die InsertAll-Methode unterscheidet sich von der Verarbeitung der Daten nicht von der insert-Methode. Über übergebene Collection wird sequentiell abgearbeitet und die einzelnen Einträge an die insert-Methode weitergereicht.</p>

Aktualisieren bestehender Einträge

Methode	Beispiel
@Deprecated update	<pre>public class BestellungenAnlegen implements InsertRule { public void insert(InsertContext context) throws BusinessException { Artikel myArticel = QueryProvider.getById(Artikel.class, 40465351L); myArticel.setEinstandspreis(3.99d); myArticel.save(); } }</pre> <p>Erläuterung:</p> <p>Der Artikel wird mit Hilfe des QueryProviders ermittelt und verändert. Im oberen Beispiel wird der Einstandspreis eines Artikels neu geschrieben. Da der Artikel bereits in der Datenbank vorhanden ist, kann mit dem Aufruf der update-Methode der Eintrag aktualisiert werden.</p>
updateAll	Die UpdateAll-Methode unterscheidet sich von der Verarbeitung der Daten nicht von der update-Methode. Über übergebene Collection wird sequentiell abgearbeitet und die einzelnen Einträge an die update-Methode weitergereicht.

Löschen bestehender Einträge

Methode	Beispiel
@Deprecated delete	<pre>public class Bestellpositionanlegen implements InsertRule { public void insert(InsertContext context) throws BusinessException { Artikel myArticel = QueryProvider.getById(Artikel.class, 40465351L); // myArticel.delete(); // oder noch falls die ID bekannt noch einfacher und schneller: Artikel.delete(40465351L); } }</pre> <p>Erläuterung:</p> <p>Ein bestehender Artikel wird aus der Datenbank gelesen, In diesem Fall ist die Id bekannt. Der Artikel kann aber auch mittels Query aus der Datenbank ermittelt werden. Anschließend wird der Eintrag physikalisch aus der Datenbank gelöscht. Anzumerken ist, dass das Löschen ein Standardprozess in Nuclos ist. Sollten noch Verknüpfungen zwischen diesem Artikel und anderen Businessobjekten bestehen, kann der Eintrag nicht gelöscht werden. Es tritt ein Fehler auf, der zum Abbruch der gesamten Regelverarbeitung führt.</p>
deleteAll	<p>Die deleteAll-Methode unterscheidet sich von der Verarbeitung der Daten nicht von der delete-Methode. Die übergebene Collection wird sequentiell abgearbeitet und die einzelnen Einträge an die delete-Methode weitergereicht.</p>
deleteLogical	<pre>public class Bestellpositionanlegen implements InsertRule { public void insert(InsertContext context) throws BusinessException { Artikel myArticel = QueryProvider.getById(Artikel.class, 40465351L); BusinessObjectProvider.deleteLogical(myArticel); } }</pre> <p>Erläuterung:</p> <p>Ein bestehender Artikel wird aus der Datenbank gelesen, In diesem Fall ist die Id bekannt. Der Artikel kann aber auch mittels Query aus der Datenbank ermittelt werden. Anschließend wird der Eintrag logisch aus der Datenbank gelöscht. Anzumerken ist, dass das Löschen ein Standardprozess in Nuclos ist. Sollten noch Verknüpfungen zwischen diesem Artikel und anderen Businessobjekten bestehen, kann der Eintrag nicht gelöscht werden. Es tritt ein Fehler auf, der zum Abbruch der gesamten Regelverarbeitung führt.</p>
deleteLogicalAll	<p>Die deleteLogicalAll-Methode unterscheidet sich von der Verarbeitung der Daten nicht von der deleteLogical-Methode. Die übergebene Collection wird sequentiell abgearbeitet und die einzelnen Einträge an die deleteLogical-Methode weitergereicht.</p>