DatasourceProvider (Beispiele)

Ausführen eines Datenquellen-Abfrage

⚠️ Achtung: Diese Anleitung funktioniert für Oracle nur für Datenquellen, die keine Daten schreiben. Dort ist es nicht erlaubt innerhalb eines SELECTs Daten zu verändern.

Methode	Beispiel

run	<pre><code>public class AuftragAbschliessen implements UpdateFinalRule {
	    public void updateFinal(UpdateContext context) throws BusinessException {
	        // Als Parameter geben wir die ID mit
	        Map<String, Object> map = new HashMap<String, Object>();
	        map.put("intid", er.getId().intValue());
	
	                DatasourceResult run = DatasourceProvider.run(AuftragDS.class, map);
	
	        for (Object[] rowWithColumns : run.getRows()) {
	            for (int idx=0; idx < run.getColumns().size(); idx++) {
	                DatasourceColumn datasourceColumn = run.getColumns().get(idx);
	                ctx.log("Feld: " + datasourceColumn.getName() + " hat den Wert: " +
	                        run.getColumns().get(idx).getType().cast(rowWithColumns[idx]).toString());
	            }
	        }
	}</code></pre>
	Erläuterung:
	
	Zur Ermittlung aller Aufträge wird im ersten Schritt eine Map mit den Parametern angelegt und dann der Datasource zur Abfrage übergeben. Das erhaltene Result-Object ist eine Liste von Einträgen, die jeder für sich ein Array aus Objekten beinhaltet. Diese Arrays beinhaltet alle Spalten des jeweiligen Eintrages mit den entsprechenden Werten. Mit Hilfe der Columns kann weiterhin ermittelt werden, um welchen Typ es sich bei der entsprechenden Spalte handelt. So kann ein - wenn gewünscht - ein Cast vorgenommen werden.
	
	Im Falle eines Laufzeit-Fehlers wird eine BusinessException geworfen.

Aufruf einer Datenbankfunktion

Methode	Beispiel

run	Der DatasourceProvider lässt sich auch nutzen, um Datenbankfunktionen aufzurufen (z.B. als Ersatz für callDBFunction der alten Nuclos API vor 4.0). Der Aufruf der Funktion selbst muss in einer Datenquelle implementiert werden, z.B.:
	
	Datenquelle: MyDatasource
	
	SQL-Statement der Datenquelle
	
	`SELECT CA_MYDBFUNCTION('$parameter')::numeric(9,2) "result"`
	
	Indirekter Aufruf der Datenbankfunktion mithilfe des DatasourceProviders.
	
	`package org.nuclet.test;`
	
	`import org.nuclos.api.exception.BusinessException;`
	`import org.nuclos.api.provider.DatasourceProvider;`
	`import org.nuclos.api.datasource.DatasourceResult;`
	
	`@Rule(name="Test", description="Test")`
	`public class Test {`
	`  public static Double callFunction() throws BusinessException {`
	`  DatasourceResult ds = DatasourceProvider.run(MyDatasourceDS.class);`
	`  Double result = (Double)(ds.getRows().iterator().next()[0]);`
	`  return result;`
	`  }`
	`}`
	
	Weitere Implementierungen von DatasourceProvider.run() lassen auch die Übergabe von Parametern an die Datenquelle (und damit an die Datenbankfunktion) zu.