

Definitionen anlegen, speichern und löschen

Die Seitenleiste

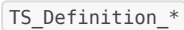
Links stehen alle Definitionen, die der geöffnete Ordner zeigt, gruppiert nach **Tabellen** und **Views**. Jeder Eintrag zeigt unter dem Namen die SQL-Tabelle, die Verbindung und – bei einer selbst angelegten – den Zusatz *eigene*.

- **Filtern ...** sucht in Name, Tabellename und Verbindung.
- Die Auswahl daneben schränkt auf *nur Tabellen* oder *nur Views* ein.
- Ein  hinter dem Namen heißt: das gespeicherte JSON dieser Definition lässt sich nicht lesen. Beim Öffnen erscheint dann der Dialog *Definition nicht lesbar* mit dem Rohtext und der Fehlermeldung – dort reparieren und **Übernehmen**; geschrieben wird erst beim Speichern.
- Die Fußzeile zählt, wie viele Einträge der Filter übrig lässt.

Aktualisieren in der Kopfleiste holt die Übersicht neu vom Server.

Neu

 legt eine Definition zu einer **vorhandenen** Tabelle an. Gibt es die Tabelle noch nicht, ist [SQL Tabelle anlegen](#) der Schritt davor.

Feld	Bedeutung
Art	<i>Tabelle (lesen und schreiben) oder View (nur lesen).</i> Bestimmt Name und Typ der Custom Property:  oder 
Name	der öffentliche Name; er steht in der Route und im Namen der Custom Property. Buchstaben, Ziffern und  , beginnend mit einem Buchstaben
Verbindung	DEXPRO-Datenbankverbindung aus der DEXPRO-Lösung
Tabelle	Tippen sucht in der Tabellenliste der Verbindung, die Treffer stehen darunter. Übernommen wird nur ein gewählter Eintrag, nicht der getippte Text

Ein Name, den es in dieser Art schon gibt, wird sofort abgewiesen – nicht erst beim Speichern, wo die Spalten schon angelegt wären.

Nach **Anlegen** steht die neue Definition im Reiter **Spalten**, noch ohne eine einzige Spalte. Der übliche nächste Griff ist dort **+ aus Tabelle ...**.

Speichern

Speichern schreibt die Definition als Custom Property und leert den Cache des TableService. Die Statuszeile nennt den Namen und die Länge des geschriebenen JSON.

- **Umbenennen:** ein geänderter Name verschiebt die Custom Property.
- **Namenskonflikt:** gibt es den Namen schon, kommt die Rückfrage *Name schon vergeben*. **Überschreiben** ersetzt den Inhalt. Wer zurückkönnen will, exportiert vorher.
- **Hinweise des Servers:** beim Schreiben räumt das Portalskript auf (leere Spalteneinträge, `schema/catalog`) und meldet Ungereimtheiten, die es nicht abweist. Sie stehen danach im Meldungsblock über dem Formular.
- **Spur einer Änderung:** wird inhaltlich wirklich etwas geändert, bekommt die Definition die Kennung `isCustomChanged`. Sie befindet sich dann nicht mehr im Zustand der Auslieferung, und der Editor sagt das beim nächsten Öffnen. Ein Speichern ohne inhaltliche Änderung setzt die Kennung nicht.

Solange etwas ungespeichert ist, steht oben das Abzeichen **nicht gespeichert**. **Verwerfen** setzt auf den zuletzt gespeicherten Stand zurück. Jeder Weg, der die geöffnete Definition verlassen würde, fragt vorher nach.

Mehr ? Als Kopie speichern ...

Schreibt den aktuellen Stand unter einem neuen Namen in eine **neue** Custom Property. Die geöffnete Definition bleibt, wie sie ist; danach ist die Kopie geöffnet.

Mehr ? Löschen

Löscht die Custom Property der Definition. Danach endet jeder Aufruf von `/<Art>/<Name>` mit **404**.

Die Definitionsdatei auf der Platte bleibt dabei unangetastet: ein erneuter Lauf von `DEXPRO__CreateCustomPropertiesFromAllTableServices` legt eine ausgelieferte Definition also wieder an. Für eine eigene gilt das nicht – sie ist weg.

Mehr ? Exportieren

Lädt die geöffnete Definition als Datei `TS_Definition_<Name>.json` bzw.

`TS_ViewDefinition_<Name>.json` herunter. Siehe auch [JSON importieren und exportieren](#).

Revision #3

Created 2026-09-17 11:03:53 UTC by Helge Czerwinski

Updated 2026-09-17 11:06:07 UTC by Helge Czerwinski