

Einstieg

- [Überblick und Aufruf](#)
- [Typischer Ablauf: von der Tabelle bis zum Ordner](#)

Überblick und Aufruf

Der **TableService-Definitionseditor** ist die Oberfläche für die Custom Properties `TS_Definition_*` und `TS_ViewDefinition_*`. Aus ihnen liest der TableServiceRouter, welche Tabellen und Sichten er über die API anbietet, welche Spalten er liefert und welche davon geschrieben werden dürfen.

Aufruf

Im Bereich **Stammdaten** stehen dafür zwei Ordner:

Ordner	Was in der Seitenleiste steht
Stammdaten-Konfiguration	die Definitionen der Stammdaten-Datenbank
Stammdaten-Konfiguration (Custom)	nur die eigenen Definitionen – die hier angelegten und die aus dem <code>Custom</code> -Ordner des TableService importierten

Beide Ordner öffnen dieselbe Oberfläche und können dasselbe. Sie unterscheiden sich allein darin, **welche** Definitionen in der Seitenleiste zur Auswahl stehen; in *(Custom)* trägt jeder Eintrag den Zusatz *eigene*.

Welche Definitionen ein Ordner zeigt, ist am Ordner hinterlegt. Weitere Ordner – etwa für eine andere Datenbank – richtet die Administration ein; der Editor selbst legt einen solchen Ordner zu einer geöffneten Definition an, siehe [Ordner und Outbar](#).

Was ein Ordner anzeigt, ist dabei eine **Anzeigeentscheidung, keine Befugnis**. Was geschrieben werden darf, entscheidet in jedem Fall das Portalskript – siehe [Geschützte Datenbanken](#).

Aufbau der Oberfläche

- **Kopfleiste** – `Neu`, `SQL Tabelle anlegen`, `SQL Tabelle ändern`, `Aktualisieren` und das Menü `Mehr` mit den Werkzeugen für Squeeze, den Tabellenaufbau und JSON.
- **Seitenleiste** – alle sichtbaren Definitionen, nach Tabellen und Views gruppiert, mit Filterfeld und Art-Auswahl.
- **Hauptbereich** – die geöffnete Definition in drei Reitern: [Allgemein](#), [Spalten](#) und [JSON](#); darüber `Speichern`, `Verwerfen` und das Menü `Mehr` der Definition.
- **Statuszeile** ganz unten – die Rückmeldung zur letzten Aktion. Grün heißt erledigt, gelb ein Hinweis, rot ein Fehler.

Backend

Jeder lesende und schreibende Aufruf geht an das Portalskript

`DEXPRO__TableServiceDefinitionEditor`. Dort fällt auch die Entscheidung, was geschrieben werden darf – siehe [Geschützte Datenbanken](#).

Typischer Ablauf: von der Tabelle bis zum Ordner

Der Weg, den eine neue eigene Tabelle im Regelfall nimmt. Jeder Schritt hat eine eigene Seite mit den Einzelheiten.

1. **Tabelle anlegen** – [SQL Tabelle anlegen](#). Verbindung wählen, Namen vergeben, Spalten eintragen. `ID` und `Licence` kommen von selbst dazu. Mit dem Haken *Definition dazu anlegen* führt der Dialog gleich zu Schritt 2.
2. **Definition anlegen** – [Definitionen verwalten](#). Art, Name, Verbindung, Tabelle.
3. **Spalten deklarieren** – [Reiter „Spalten“](#). + **aus Tabelle ...** übernimmt die Spalten der SQL-Tabelle samt Vorschlägen für Typ, Schlüssel- und Mandantenspalte. Was hier nicht steht, liefert die API nicht.
4. **Feinschliff** – [Reiter „Allgemein“](#) für Sortierung und Filter, im Reiter *Spalten* die Beschriftungen und Wertelisten.
5. **Meldungen lesen** – [Prüfungen und Meldungen](#). Fehlt der Primärschlüssel oder eine NOT-NULL-Spalte, steht das dort, bevor es im Client auffällt.
6. **Speichern**. Danach ist die Definition über die API erreichbar.
7. **Ordner anlegen** – [Ordner und Outbar](#), damit die Tabelle im Client erscheint. Danach ab- und wieder anmelden.

Dazu kommen die Wege, die man später braucht:

- Eine Spalte fehlt → [SQL Tabelle ändern](#); der Dialog zieht die betroffenen Definitionen gleich nach.
- Der Stand soll zum Kunden → [Alle SQL-Tabellen exportieren / importieren](#) für den Tabellenaufbau und [Definitionen importieren und exportieren](#) für die Definitionen.
- Die Tabelle soll nach Squeeze → [Tabelle in Squeeze Stammdaten hinzufügen](#), oder umgekehrt [Squeeze-Stammdaten in Tabelle laden](#).