SimplexScenarioEditor

1. Einleitung

Wie im Kapitel zum Datenmodell formuliert, beruht Simplex4Data u.a. auf dem Architekturprinzip, Daten atomar in einer simplen Struktur zu speichern.

Auf diesen simplen Strukturen können beliebige (und beliebig komplexe) Zusammenstellungen aufsetzen. Diese Zusammenstellungen werden in Simplex als **Szenarien** bezeichnet. Ein Szenario kann aus mehreren **Sichten** bestehen, wobei jede Sicht eine in sich abgeschlossene Datensammlung ist. das wird in Analogie zu Tabellenkalkulationsprogrammen verständlicher: Das Szenario ist das ganze Dokument, die Sichten sind die einzelnen Tabellen / Blätter des Dokuments. Häufig sind die Views technisch gesehen Datenbank-Views.

Szenarien werden auf unterschiedliche Art und Weise erzeugt:

- Das Standardszenario wird durch interne Prozesse des Simplex4Data erzeugt und bildet stets den aktuellen Stand des Datenpools ab.
- Andere Szenarios werden durch Expert*innen implementiert, um fest definierte Standardformate zu erfüllen (z.B. INSPIRE).
- Wieder andere werden - mittels des SimplexScenarioEditor - zusammengestellt, um ad hoc aufkommende Fragestellungen zu klären. Sie können von jeder Person, die Zugriff auf das Webinterface hat, erzeugt werden.

Im Weiteren wird zunächst das Standardszenario 0 vorgestellt.

2. Das Standardszenario 0

Das Szenario 0 wird durch interne Prozesse des Simplex4Data erzeugt. Wurde das Simplex4Data korrekt installiert und bedient (das bedeutet insbesondere, dass das Metadatensystem intakt ist und die Datenbank nur durch Funktionen des Simplex4Data manipuliert wird), dann sind diese Sichten immer verfügbar. Sie werden innerhalb des Simplex4Data selbst vielfältig verwendet und bilden auch die Basis der Datenbereitstellung über Dienste.

2.1. oViews

Zu jeder definierten Klasse wird automatisch eine Sicht erzeugt, in welcher die Objekte der Klasse mit ihren Standardfeldern (weiß) mit allen definierten Sachattributen (blau) zusammengespielt werden. Diese Sicht wird als „oView“, kurz für „Objekt-View“, bezeichnet. Die oView passt sich dynamisch an Änderungen der Datenhaltung wie beispielsweise der Definition weiterer Sachattribute, Umbenennungen, Löschungen, etc. an.

Die oViews werden im Szenario 0 über den SimplexService als collections bereitgestellt. Sie sind demnach auch der Kern der Datenbereitstellung nach außen.

Die Abbildung zeigt einen Ausschnitt* des oViews der Klasse Gemeinden. Zu sehen sind einige Standardfelder (weiß) und einigen Sachattribute (blau, hervorgehoben), die in den Tutorials zur Definition

und Konvertierung von Sachattributen behandelt wurden.

*Es handelt sich um einen Ausschnitt sowohl in der Länge als auch in der Breite der Tabelle: einige interne Felder sowie Standardfelder wurden ausgespart, die Tabelle ist also deutlich breiter als die hier sichtbaren Spalten es vermuten lassen.

Data Output | Explain | Messages | Notifications

	o integer	c integer	nam character varying (255)	typ character varying (255)	ndx character varying (255)	flaeche double precision	bev_insgesamt integer	postleitzahl text	pt geometry
1	319404	100	Flensburg	Stadt	010010000000	56.73	91113	24937	0101000020E6100...
2	319405	100	Kiel	Stadt	010020000000	118.65	246243	24103	0101000020E6100...
3	319406	100	Lübeck	Stadt	010030000000	214.19	216277	23539	0101000020E6100...
4	319407	100	Neumünster	Stadt	010040000000	71.66	79496	24534	0101000020E6100...
5	319408	100	Brunsbüttel	Stadt	010510011011	65.21	12381	25541	0101000020E6100...
6	319409	100	Heide	Stadt	010510044044	31.97	21844	25746	0101000020E6100...
7	319410	100	Averlak	Gemeinde	010515163003	9.06	558	25715	0101000020E6100...
8	319411	100	Brickeln	Gemeinde	010515163010	6.07	198	25712	0101000020E6100...
9	319412	100	Eddelak	Gemeinde	010515163024	9.21	1345	25715	0101000020E6100...
10	319413	100	Eggstedt	Gemeinde	010515163026	12.79	758	25721	0101000020E6100...

2.2. yViews

Analog zum oView für jede Klasse wird für jede Verbindung eine sogenannte yView generiert. Während die Verbindungstabelle nur den notwendigen Kern der reinen Verbindungsinformation speichert (vgl. Visualisierung in der Seite zur Konvertierung von Verbindungen), ergänzt die yView diese Information um die Standardfelder der beiden jeweils verbundenen Objekte. Auf diese Weise sind auf einen Blick zahlreiche wichtige Informationen zu erkennen.

Die Abbildung zeigt einen Ausschnitt* der yView zur Verbindung Kreise -> Gemeinden. Damit wird auf einen Blick sichtbar, welche Gemeinde zu welchem Landkreis gehört.

*Es handelt sich um einen Ausschnitt sowohl in der Länge als auch in der Breite der Tabelle: Es wurden gezielt einzelne Spalten selektiert.

Data Output | Explain | Messages | Notifications

	typ1 character varying (255)	nam1 character varying (255)	o1 integer	o2 integer	typ2 character varying (255)	nam2 character varying (255)
1	Kreis	Soest	5743	322271	Gemeinde	Möhnesee
2	Kreis	Kleve	5702	321737	Stadt	Goch
3	Landkreis	Bad Kissingen	5925	327843	Gemeindefreies Gebiet	Waldfensterer Forst
4	Landkreis	Rostock	5975	329099	Gemeinde	Reddelich
5	Landkreis	Görlitz	5987	327543	Gemeinde	Schönau-Berzdorf a.d. E...
6	Landkreis	Kaiserslautern	5801	325540	Stadt	Landstuhl
7	Landkreis	Rhein-Pfalz-Kreis	5804	325737	Gemeinde	Hochdorf-Assenheim
8	Landkreis	Rottweil	5836	326582	Gemeinde	Deißlingen
9	Landkreis	Hersfeld-Rotenburg	5766	323204	Gemeinde	Hohenroda
10	Landkreis	Aichach-Friedberg	5937	328079	Gemeinde	Mering

2.3. geoViews

Diese Views werden speziell angelegt, um eine Datenbereitstellung mittels des GeoServer zu gewährleisten. Diese Sichten sind in einem eigenen Datenbankschema namens "simplefeatures", zu finden.

Der Name dieses Schemas ist Programm: Die Sichten übertragen die Klassen (bzw. oViews) des Simplex4Data in Layer, die dem OGC Simple Feature Access Modell entsprechen. Da dieses gegenüber dem Datenmodell des Simplex4Data erhebliche Unterschiede im Umgang mit Geometrien aufweist, ist dieses eigene Schema und die darin angelegten Views notwendig, um eine "Übersetzung" in die Welt des OGC Simple Features zu gewährleisten.

Die wesentlichen Unterschiede im Umgang mit Geometrien sind hier kurz zusammengefasst:

OGC Simple Feature Modell	Simplex4Data Datenmodell
Features in Layern	Objekte in Klassen
Geometrie ist das objektbildende Kriterium.	Geometrie ist ein Sachattribut unter vielen.
Ein Feature hat immer genau eine Geometrie.	Ein Objekt kann 0-N Geometrien aufweisen.
Ein Layer besteht aus Features, die alle Geometrien desselben Typs (z.B. Point) aufweisen.	Eine Klasse kann Objekte enthalten, von denen jedes einzelne 0-N Geometrien verschiedenen Geometrietyps aufweisen kann.

Häufig erfordert die Übersetzung es, aus einer Klasse des Simplex4Data mehrere Layer zu erzeugen. Aus einer Klasse Gemeinde werden dann z.B. folgende geoViews:

- Gemeinden_Mittelpunkt_Punkt,
- Gemeinden_Administrationssitz_Punkt,
- Gemeinden_Flaeche_Polygon.

Aus diesen geoViews bezieht der optionale GeoServer-Container des Simplex4Data seine Daten und stellt sie (je nach Konfiguration) als WFS und/oder WMS bereit.

3. Händisches Anlegen von Szenarien (Prototyp)

Der SimplexScenarioEditor ist noch in einer prototypischen Phase, sein Interface ist noch starken Änderungen unterworfen. Deshalb kann hier noch keine hilfreiche Dokumentation erfolgen.

- SimplexVisual
- Nach oben