

Stand 01.01.2023

Technische Dokumentation zur Spezifikation

Datenauswertungsstelle nach § 113 Abs. 1b SGB XI (DAS Pflege)

V03.0

Impressum

Herausgeber

Unabhängige Datenauswertungsstelle nach § 113 Abs. 1b SGB XI:
aQua – Institut für angewandte Qualitätsförderung und Forschung im Gesundheitswesen GmbH

Thema

Technische Dokumentation zur Spezifikation

Auftraggeber

Qualitätsausschuss Pflege nach § 113b SGB XI

Hinweis

Aus Gründen der leichten Lesbarkeit wird im Folgenden auf eine geschlechtsspezifische Differenzierung verzichtet. Entsprechende Begriffe gelten im Sinne der Gleichbehandlung für alle Geschlechter.

Anschrift des Herausgebers

aQua – Institut für angewandte Qualitätsförderung
und Forschung im Gesundheitswesen GmbH
Maschmühlenweg 8–10 · 37073 Göttingen

Telefon (+49) 0551-789 52-282
Telefax (+49) 0551-789 52-10

support-swa@das-pflege.de

www.das-pflege.de
www.aqua-institut.de

Zertifiziert nach ISO 9001:2015
und ISO/IEC 27001:2013

Inhaltsverzeichnis

| | | |
|----------|--|-----------|
| 1 | Hintergrund | 6 |
| 1.1 | Spezifikationsbegriff | 6 |
| 1.2 | Zielgruppe und Zielsetzung der Technischen Dokumentation | 6 |
| 1.3 | Komponenten | 6 |
| 1.4 | Veröffentlichung neuer Versionen | 7 |
| 1.5 | Versionierung | 7 |
| 2 | Datenerhebung | 7 |
| 2.1 | Zeiträume und Fristen | 7 |
| 2.2 | Datenerhebung | 8 |
| 3 | Beschreibung des XML-Schema (Syntax) | 10 |
| 3.1 | Strukturelemente (Compositors) | 10 |
| 3.2 | Kardinalitäten (Cardinality Constraints) | 10 |
| 3.3 | Komplexe Datentypen | 10 |
| 4 | XML-Schema (interface_qs_data) | 11 |
| 4.1 | Element <root> | 12 |
| 4.2 | Element <header> | 13 |
| 4.2.1 | Element <header>/<document> | 14 |
| 4.2.2 | Element <header>/<care_provider> | 15 |
| 4.2.3 | Element <header>/<software> | 16 |
| 4.3 | Element <body> | 16 |
| 4.3.1 | Element <body>/<commentation_container> | 17 |
| 4.3.2 | Element <body>/<data_container>/<facility> | 18 |
| 4.3.3 | Element <body>/<data_container>/<residents> | 19 |
| 4.4 | Element <qs_data> | 20 |
| 4.5 | Definition von Datenfeldern | 21 |
| 4.5.1 | Einschränkungen (Facetten) | 21 |
| 4.5.2 | Muss/Kann-Felder (required und optional) | 22 |
| 4.5.3 | Listenfelder | 23 |
| 4.5.4 | Zusatzinformationen | 23 |
| 4.5.5 | Abhängigkeiten von Datenfeldern - Feldgruppen | 24 |
| 4.6 | Schemavalidierung | 24 |
| 4.6.1 | Regeln innerhalb des XML-Schemas (fachliche Fehler) | 24 |
| 4.6.2 | Regeln außerhalb des XML-Schemas (technische Fehler) | 25 |
| 4.6.3 | Regelarten | 26 |
| 4.7 | Validierungsprotokoll | 26 |

| | | |
|-----|---|----|
| 5 | Tabellarische Darstellung der Zeiträume und Fristen | 29 |
| 6 | Webservice (REST-API) | 30 |
| 6.1 | Client Implementierungen | 30 |
| 6.2 | Testumgebung | 30 |
| 6.3 | REST-API-Dokumentation | 31 |

Tabellenverzeichnis

| | |
|---|----|
| Tabelle 1: Strukturelemente (Compositors) | 10 |
| Tabelle 2: Kardinalitäten (Cardinality Constraints) | 10 |
| Tabelle 3: Beschreibung der Schema-Dateien | 11 |
| Tabelle 4: Beschreibung der Kind-Elemente von <document>..... | 14 |
| Tabelle 5: Beschreibung der Kind-Elemente von <care_provider>..... | 15 |
| Tabelle 6: Beschreibung der Kind-Elemente von <software> | 16 |
| Tabelle 7: Beschreibung der Kind-Elemente von <residents> | 19 |
| Tabelle 8: Facetten | 22 |
| Tabelle 9: Datenfeldspezifische Zusatzinformationen in <xs:appinfo> | 23 |
| Tabelle 10: Regelspezifische Zusatzinformationen in <xs:appinfo> | 25 |
| Tabelle 11: Regelarten | 26 |
| Tabelle 12: CSV-Format - Zeiträume und Fristen | 29 |
| Tabelle 13: Matrix - Personen- vs. Maschinenzugang | 30 |

Abbildungsverzeichnis

| | |
|---|----|
| Abbildung 1: Zeiträume und Fristen | 7 |
| Abbildung 2: Schema-Dateien..... | 11 |
| Abbildung 3: Element <root> mit Kind-Elementen | 12 |
| Abbildung 4: Element <header> mit Kind-Elementen | 13 |
| Abbildung 5: Element <document> mit Kind-Elementen | 14 |
| Abbildung 6: Element <care_provider> mit Kind-Elementen | 15 |
| Abbildung 7: Element <software> mit Kind-Elementen | 16 |
| Abbildung 8: Element <body> mit Kind-Elementen..... | 17 |
| Abbildung 9: Element <commentation_container> mit Kind-Elementen..... | 17 |
| Abbildung 10: Element <facility> mit Kind-Elementen | 18 |
| Abbildung 11: Element <residents> mit Kind-Elementen | 19 |
| Abbildung 12: Element <validation_status> mit Kind-Elementen | 26 |
| Abbildung 13: Element <delivery_status> mit Kind-Elementen | 27 |

1 Hintergrund

Die fachlich unabhängige Datenauswertungsstelle nach § 113 Abs. 1b SGB XI (DAS Pflege) nimmt von allen dokumentationspflichtigen vollstationären Pflegeeinrichtungen zweimal jährlich pseudonymisierte Daten von Bewohnern entgegen, die zu den einrichtungsindividuellen Stichtagen in der Einrichtung leben. Auf Basis dieser Daten erfolgt die Berechnung der statistischen Plausibilitätskontrolle und Qualitätsindikatoren anhand festgelegter Rechenregeln und eine Einstufung der Ergebnisse. Die rechtlichen Grundlagen bilden die Maßstäbe und Grundsätze für die Qualität, die Qualitätssicherung und -darstellung sowie für die Entwicklung eines einrichtungsinternen Qualitätsmanagements nach § 113 SGB XI in der vollstationären Pflege (Maßstäbe und Grundsätze) und die Vereinbarung nach § 115 Abs. 1a SGB XI über die Darstellung und Bewertung der Qualitätsindikatoren gemäß § 113 Abs. 1a SGB XI und der Ergebnisse aus Qualitätsprüfungen nach §§ 114 f. SGB XI (QDVS).

1.1 Spezifikationsbegriff

Die Spezifikation ist die Gesamtheit aller Vorgaben, nach denen die Dokumentation der QS-Daten auf Seiten der Anwender erfolgen muss. Zudem werden Festlegungen in Bezug auf die Datenübermittlung an die DAS Pflege getroffen. Sie ist ein verbindliches Regelwerk für alle am Projekt Beteiligten, um die technische Umsetzung inklusive der Erfüllung der Datenschutz- und Datensicherheitsanforderungen zu gewährleisten. Softwareanbieter finden darin alle notwendigen Informationen, um ggf. bereits bestehende Dokumentationssoftware dahingehend zu entwickeln, dass eine Dokumentation und Datenübertragung aus den bestehenden Systemen möglich werden. Die Spezifikation bildet zusätzlich die Grundlage für die Datenerhebung über das Portal der DAS Pflege, sofern keine externe Software zum Einsatz kommt.

1.2 Zielgruppe und Zielsetzung der Technischen Dokumentation

Die Technische Dokumentation ist Bestandteil der Spezifikation und so ausgestaltet, dass sie von QS- und/oder IT-/EDV-Experten verstanden wird. Sie richtet sich primär an diesen Teilnehmerkreis. Mit der Technischen Dokumentation wird das Ziel verfolgt, die Komponenten der Spezifikation umfassend und verständlich zu beschreiben und damit die Entwicklung von Systemen auf Basis der Spezifikation zu ermöglichen. In Bezug auf die Gestaltung der Softwareprodukte und die Wahl der Technologien bzw. Programmiersprachen werden keine Vorgaben gemacht. Jedoch sind die in dieser Spezifikation festgelegten Vorgaben in Bezug auf Datensatzbeschreibungen, Datenprüfungen und Datenformate verbindlich.

Die Datenübertragung an die DAS Pflege erfolgt ausschließlich in Form von XML-Dokumenten. Struktur und Inhalt werden in Form von XML-Schemata (XSD) in der Version 1.1 beschrieben. Die in dieser Spezifikation definierten Datensätze basieren auf dem in Anlage 3 der Maßstäbe und Grundsätze dargestellten Erhebungsinstrument, können jedoch in Ausnahmefällen davon abweichen. Verbindlich sind ausschließlich die Vorgaben dieser Spezifikation.

1.3 Komponenten

Die Spezifikation in der Gesamtheit umfasst die folgend aufgeführten Komponenten:

- Übersicht der Anpassungen/Änderungen im Vergleich zur Vorversion
- Technische Dokumentation zur Spezifikation (pdf; dieses Dokument)
- Datensatzbeschreibungen als XML-Schema (xsd)
- Muster-Dokumentationsbogen zur Visualisierung der Datensätze (pdf)
- Ausfüllhinweise zu einzelnen Datenfeldern (pdf)
- Tabellarische Darstellung der verbindlichen Fristen für die Datenübertragung auf Basis des einrichtungsindividuell gewählten Stichtags (csv)

1.4 Veröffentlichung neuer Versionen

In Absprache zwischen dem Qualitätsausschuss Pflege und der DAS Pflege wurden verbindliche Termine für die Veröffentlichung neuer Spezifikationsversionen bzw. Releases festgelegt. Nähere Informationen finden sich auf den Webseiten der DAS Pflege unter <https://www.das-pflege.de/spezifikationen>.

1.5 Versionierung

Die Spezifikation wird auf der Webseite der DAS Pflege als zip komprimierter Ordner zum Download kostenfrei zur Verfügung gestellt.

Das Spezifikationspaket ist immer nach der folgenden Konvention benannt:

DAS_Pflege_Spezifikation_V<Versionsnummer>.<Release>

2 Datenerhebung

Eine Kernaufgabe der DAS Pflege besteht in der Annahme, Prüfung und Auswertung der Qualitätssicherungsdaten. Die Datenerfassung über das Webportal oder die Datenübermittlung per Dateiupload oder REST-Schnittstelle erfolgt zweimal jährlich durch jede Pflegeeinrichtung und ist für jede Einrichtung an individuelle Fristen und Zeiträume gebunden. Außerhalb dieser Zeiträume werden keine Datensätze der jeweiligen Pflegeeinrichtung von der DAS Pflege bzw. durch den Datenservice angenommen.

2.1 Zeiträume und Fristen

Die folgende Abbildung zeigt, in Anlehnung an § 4 Anlage 1 der Maßstäbe und Grundsätze und Abbildung 1 in Anlage 3 der Maßstäbe und Grundsätze, die Zeiträume eines kompletten Zyklus. Im Rahmen der Spezifikation wird der Zeitraum vom Beginn des Erhebungszeitraums (erster Tag nach dem vorherigen Stichtag) bis hin zum Abschluss des Kommentierungszeitraums unter dem Begriff „Zyklus“ zusammengefasst.

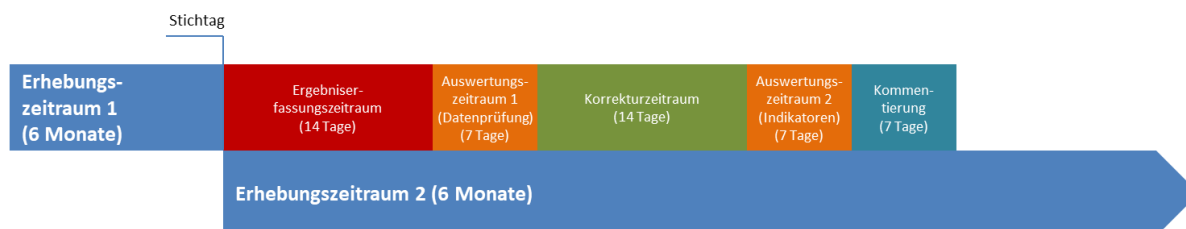


Abbildung 1: Zeiträume und Fristen

Jede Pflegeeinrichtung legt während des Registrierungsprozesses eine Stichtagskombination zunächst ohne Jahresbezug fest. Der Abstand zwischen den Stichtagen beträgt 6 Monate, sodass jeweils ein Stichtag in der ersten und zweiten Jahreshälfte liegt (bspw. die Kombination 15.03. und 15.09.). Die zu diesem Zeitpunkt festgelegte Stichtagskombination ist fortan unveränderlich. Die Stichtage und Fristenden (letzter Tag eines Zeitraums) dürfen gemäß den Festlegungen nicht auf ein Wochenende oder einen gesetzlichen Feiertag fallen. In diesen Fällen werden die Daten immer auf den nächsten folgenden Werktag verschoben, wobei der Samstag nicht als Werktag gilt. Auf Seiten der DAS Pflege werden auf Basis der Stichtagskombinationen alle folgenden Stichtage und Fristen für die Pflegeeinrichtungen anhand der dargestellten Regelung ermittelt und sukzessive erweitert.

- **Erhebungszeitraum (ca. 6 Monate):** Beschreibt den Zeitraum nach dem vorherigen Stichtag (+ 1 Tag) bis einschließlich dem aktuellen Stichtag. Mit Beginn dieses Zeitraums können Daten zum aktuellen Zyklus an die DAS Pflege übermittelt werden. Auf Grund der Verschiebungen variiert die Dauer des Zeitraums in der Regel zwischen 178 bis 186 Tagen.

- **Ergebniserfassungszeitraum (14 Tage):** Innerhalb dieses Zeitraums werden Daten (abschließend) dokumentiert, sofern es sich um Angaben handelt, die erst nach dem Stichtag valide zu beantworten sind und vor Ablauf an die DAS Pflege übermittelt.
- **Auswertungszeitraum 1 (7 Tage):** Berechnung der statistischen Plausibilitätskontrolle auf Seiten der DAS Pflege. Eine Datenübertragung an die DAS Pflege ist in diesem Zeitraum nicht möglich.
- **Korrekturzeitraum (14 Tage):** Auf Basis der statistischen Plausibilitätskontrolle können Einrichtungen Korrekturen vornehmen und die Daten erneut an die DAS Pflege übermitteln. Sofern keine Anpassungen erfolgen, ist eine erneute Datenübermittlung nicht notwendig. Nach Ablauf dieses Zeitraums ist eine Anpassung der QS-Daten zu Bewohnern bzw. eine erneute Datenlieferung zur aktuellen Erhebung nicht mehr möglich.
- **Auswertungszeitraum 2 (7 Tage):** Erneute Berechnung der statistischen Plausibilitätskontrolle, Berechnung der Qualitätsindikatoren und Ziehung der Stichprobe(n) gemäß den Maßstäben und Grundsätzen auf Seiten der DAS Pflege.
- **Kommentierungszeitraum (7 Tage):** Nach Erhalt der Indikatorergebnisse kann optional eine Kommentierung der Ergebnisse von Seiten der Pflegeeinrichtungen erfolgen. Die Kommentare werden zur Veröffentlichung durch die DAS Pflege weitergeleitet.

Es ist zu beachten, dass immer eine Überschneidung zwischen zwei aufeinander folgenden Zyklen besteht und ggf. Daten im gleichen Zeitraum zu verschiedenen Zyklen geliefert werden können. Das betrifft den Ergebniserfassungs- und den Korrekturzeitraum des aktuellen Zyklus, die gleichzeitig im bereits begonnen Erhebungszeitraum des folgenden Zyklus liegen. Aus diesem Grund sind Datenlieferungen durch die Angabe des Stichtags eindeutig zu referenzieren, um die korrekte Zuordnung der Daten gewährleisten zu können.



Achtung

In einer Datenlieferung sind jeweils alle Datensätze zu Bewohnern eines Erhebungszeitraums zu liefern. Bereits bei der DAS Pflege persistierte Daten werden durch eine erneute Datenlieferung derselben Pflegeeinrichtung für denselben Zyklus vollständig überschrieben. Dies ist insbesondere bei Korrekturlieferungen zu beachten. Gleiches gilt für die Übermittlung von Kommentierungen. Gültig sind demnach immer die Daten der letzten validen bzw. erfolgreichen Datenübermittlung.

2.2 Datenerhebung

Die Datenerfassung ist ausschließlich für natürliche oder technische Benutzer einer Pflegeeinrichtung möglich. D.h. es ist zwischen Personenzugängen und Maschinenzugängen in den unterschiedlichen Kommunikationskanälen zu unterscheiden.

Die DAS Pflege unterscheidet zwischen drei Erfassungsmodi:

1. **Formular:** Die Datenerfassung erfolgt mittels manueller Erfassung über ein Webformular im Webportal der DAS Pflege. Die Erfassung kann durch mehrere Benutzer gleichzeitig erfolgen (= Mehrbenutzermodus).
2. **Upload:** Die Datenerfassung erfolgt mittels manuellen Uploads einer spezifikationskonformen XML-Datei über die Upload-Funktion im Webportal der DAS Pflege.
3. **Webservice:** Die Datenerfassung erfolgt mittels automatisierter Datenübermittlung von spezifikationskonformen XML-Dateien via Schnittstelle bzw. REST-Webservice.

Eine Pflegeeinrichtung kann die Datenerfassung nur über einen der aufgeführten Erfassungsmodi durchführen. Mit der Registrierung einer Pflegeeinrichtung wird der Erfassungsmodus initial auf Formular gesetzt, d.h. die berechtigten Personen der Pflegeeinrichtung können die Daten im Mehrbenutzermodus über die Formulare des Portals erfassen (Weberfassung).

Ein Benutzer der Pflegeeinrichtung mit der Rolle Administrator kann den Erfassungsmodus im Webportal einmalig ändern. Dies ist für die folgenden Konstellationen möglich:

- Formular → Upload
- Formular → Webservice
- Upload → Webservice

Folgende Änderungen sind ausschließlich durch die DAS Pflege möglich und entsprechend Kontakt zur DAS Pflege aufzunehmen:

- Webservice → Formular
- Webservice → Upload
- Upload → Formular

Alle drei Erfassungsmodi implementieren die Spezifikation und setzen das Vorhandensein eines aktiven Zyklus mit einem konkreten Stichtag voraus. Die erfassten Daten werden unabhängig vom Kommunikationskanal stets gegen die zum Stichtag gültige Spezifikationsversion validiert. Die Ergebnisse der Validierung werden je nach Kanal wie folgt publiziert:



1. Formular: Validierung auf Datensatzebene. Die Ergebnisse (Warnungen und Fehler) werden tabellarisch unterhalb des Erfassungsformulars dargestellt. Sprungmarken und schwebende Menüelemente erleichtern die Navigation im Formular und in den Validierungsergebnissen. Sofern keine fachlichen Fehler vorliegen, erfolgt anschließend ein Statuswechsel des Formulars zu „geprüft, plausibel“, was die Voraussetzung für die Freigabe eines Datensatzes durch einen berechtigten Benutzer der Einrichtung ist. Der Prozess der Datenfreigabe ist für die anderen Modi obsolet und wird an dieser Stelle nicht näher erläutert.
2. Upload: Validierung der gesamten Importdatei. Die über den Upload-Dialog importierte XML-Datei wird im ersten Schritt auf technische Vorgaben und Lesbarkeit und im zweiten Schritt auf die fachliche Korrektheit überprüft. Nur technisch korrekte Importdateien (d.h. eine wohlgeformte XML-Datei) mit fachlich validen Datensätzen (keine Fehler vorhanden) werden in die Datenverarbeitung übergeben und persistiert. Die Validierungsergebnisse werden unmittelbar nach dem Importvorgang als eine aufeinanderfolgende Auflistung der Warnungen und Fehler dargestellt. Zusätzlich wird jeder Importvorgang vom System protokolliert. Das Protokoll beinhaltet den Zeitpunkt des Importvorganges, die originale Importdatei und das Validierungsprotokoll, welches die Implementierung der veröffentlichten Spezifikation darstellt, die Originalwerte der Importdatei beinhaltet und diese um Validierungsergebnisse ergänzt. Die Art der Fehlerausgabe ist für die maschinelle Verarbeitung gedacht.
3. Webservice: Validierung der gesamten Importdatei. Die über den REST-Webservice importierte XML-Datei wird im ersten Schritt auf technische Vorgaben und Lesbarkeit und im zweiten Schritt auf die fachliche Korrektheit überprüft. Nur technisch korrekte Importdateien (d.h. eine wohlgeformte XML-Datei) mit fachlich validen Datensätzen (keine Fehler vorhanden) werden in die Datenverarbeitung übergeben und persistiert. Jeder Importvorgang wird protokolliert. Das Protokoll beinhaltet den Zeitpunkt des Importvorganges, die originale Importdatei und das Validierungsprotokoll, welches die Implementierung der veröffentlichten Spezifikation darstellt, die Originalwerte der Importdatei beinhaltet und diese um Validierungsergebnisse ergänzt. Die Validierungsprotokolle können vom Maschinenbenutzer über einen REST-Webservice gelesen werden.

3 Beschreibung des XML-Schema (Syntax)

Struktur und Inhalt der Datensätze wird mit einem XML-Schema (XSD) in der Version 1.1 beschrieben¹. Für die grafische Darstellung der Struktur des XML-Schemas, sowie zur Darstellung von Kardinalitäten werden im Rahmen dieses Dokuments folgende Diagramme und Symbole² genutzt.



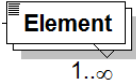
3.1 Strukturelemente (Compositors)

Tabelle 1: Strukturelemente (Compositors)

| Symbol | Beschreibung |
|---|---|
|  | xs:sequence Alle Kind-Elemente müssen in der festgelegten Reihenfolge angegeben werden. |
|  | xs:choice Genau ein Kind-Element muss ausgewählt werden. |

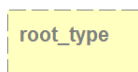
3.2 Kardinalitäten (Cardinality Constraints)

Tabelle 2: Kardinalitäten (Cardinality Constraints)

| Symbol | Kardinalität | Beschreibung |
|---|--------------|---|
|  | 1 | Obligatorisches Element Es muss genau einmal vorkommen. |
|  | 0..1 | Optionales Element Es kann keinmal oder genau einmal vorkommen. |
|  | n..m | Wiederholendes Element Es muss genau n-mal, aber darf nur maximal m-mal vorkommen. Im Beispiel einmal bis unbegrenzt (= unbounded). |

3.3 Komplexe Datentypen

Gelb hinterlegte und umrandete Elementgruppen stellen komplexe Datentypen `<xs:complexType>` dar. Der Name des Datentyps wird jeweils in der linken oberen Ecke ausgewiesen:



Grundsätzlich werden Dateninhalte immer über das Attribut `@value` repräsentiert, d.h. es finden sich keine Daten innerhalb von Element-Tags:

```
<ERHEBUNGSDATUM value="2024-06-20"/>
```

An geeigneten Stellen werden Inhalte innerhalb des Schemas dokumentiert. Hierfür wird das Element `<xs:annotation>` bzw. die Elemente `<xs:documentation>` und `<xs:appinfo>` zur Annotation verwendet. Im Allgemeinen sind die Namen der Elemente und Attribute in englischer Sprache definiert. Im Fall der spezifischen Datenfelder in den Datenätzen zur Dokumentation der QS-Daten wird auf eine Übersetzung auf Grund der teilweise verwendeten Fachterminologie verzichtet. In anderen Fällen werden Elemente in Bezug auf die Einheitlichkeit nicht übersetzt.

¹ <https://www.w3.org/TR/xmlschema11-1/>

² Entnommen aus dem XML-Editor XMLSpy der Firma Altova

4 XML-Schema (interface_qs_data)

In diesem XML-Schema sind die Datensätze definiert, die von Seiten der Pflegeeinrichtungen an die DAS Pflege verpflichtend oder optional (im Fall der Kommentierung) zu übermitteln sind.

Eine Datenlieferung umfasst:

- Einrichtungsbezogene Daten zur Vollzähligkeitsanalyse
- Maximal ein Element zu allen Bewohnern, die gemäß den Festlegungen in den Maßstäben und Grundsätzen dokumentationspflichtig sind. Für die entsprechenden Bewohner ist entweder ein kompletter Datensatz mit allen verpflichtenden Angaben zum Status und der Versorgung oder die Nennung eines Ausschlussgrundes für die aktuelle Erhebung zu dokumentieren.

Neben den Datensatz- und Felddefinitionen beinhaltet die Spezifikation feldbezogene und feldübergreifende Plausibilitätsregeln, sowie ergänzende Information zur Dokumentation einzelner Datenfelder (= Ausfüllhinweise).

Die Schema-Dateien sind im Ordner **interface_qs_data** zusammengefasst.

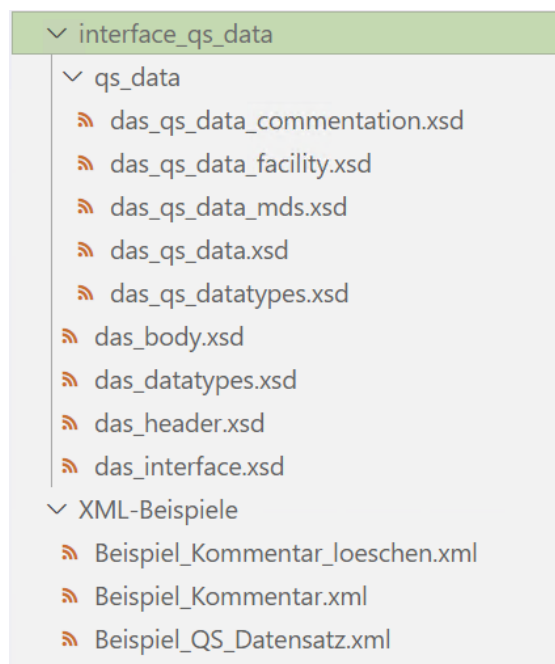


Abbildung 2: Schema-Dateien

Das Schema zur Definition der Datensätze umfasst die in der folgenden Tabelle aufgeführten Schema-Dateien.

Tabelle 3: Beschreibung der Schema-Dateien

| Datei | Kurzbeschreibung des Inhalts |
|-------------------|---|
| das_interface.xsd | Definition des <root>-Elements |
| das_header.xsd | Definition des Elements <header> als Umschlagselement für Informationen zum Dokument, zum Datenlieferanten, sowie zu dem vom Datenlieferanten verwendeten Softwareprodukt |
| das_body.xsd | Definition des Elements <body> als Umschlagselement für die QS-Daten |
| das_datatypes.xsd | Definition global verwendeter Datentypen |
| das_qs_data.xsd | Datensatz zur Dokumentation von Bewohnern ohne Ausschlussgrund |

| | |
|------------------------------|--|
| das_qs_data_mds.xsd | Datensatz zur Dokumentation von Bewohnern mit Ausschlussgrund MDS = Minimaldatensatz |
| das_qs_data_facility.xsd | Datensatz zur Erfassung einrichtungsbezogener Daten zur Vollzählkeitsanalyse |
| das_qs_data_commentation.xsd | Kommentierung |
| das_qs_datatypes.xsd | Definition der Datentypen, die in den Datensätzen verwendet werden. In der Regel handelt es sich dabei um Einschränkungen auf einen Wertebereich (<code>xs:enumeration</code>) |

4.1 Element <root>

Das Element `<root>` folgt auf die Headerzeile und stellt das Umschlagselement für alle weiteren im Schema definierten XML-Typen dar. Als Zeichensatz wird UTF-8 (Unicode-Codierung) verwendet. Dem `<root>`-Element sind keine weiteren Attribute zugewiesen. Alle Elemente unterhalb von `<root>` befinden sich im Namensraum (Namespace) `https://www.das-pflege.de` der ohne Präfix-Mapping im `<root>`-Element in das XML eingebunden wird.

Beispiel:

```
<?xml version="1.0" encoding="UTF-8"?>
<root xmlns="https://www.das-pflege.de">
... Dokumentinhalt
</root>
```

Die Kind-Elemente bestehen immer aus einer Sequenz der Elemente `<header>` und `<body>`. Die Unterscheidung in die übergeordneten Elemente `<header>` und `<body>` erfolgt mit dem Ziel die dokumentbeschreibenden administrativen Angaben von den fachlich inhaltlichen Daten abzugrenzen. Das optionale Element `<delivery_status>` wird im weiteren Verlauf des Dokumentes beschrieben.

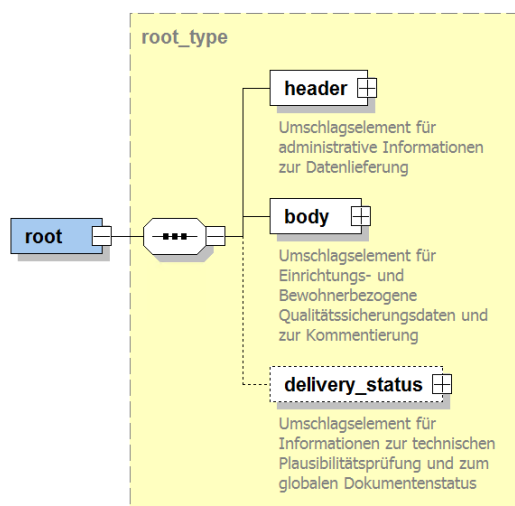


Abbildung 3: Element `<root>` mit Kind-Elementen

Die Grundstruktur des XML-Dokumentes ausgehend vom Element `<root>` stellt sich demnach wie folgt dar.

Beispiel:

```
<?xml version="1.0" encoding="UTF-8"?>
<root xmlns="https://www.das-pflege.de">
  <header>
    ... administrative Daten
  </header>
  <body>
    ... fachliche QS-Daten
  </body>
</root>
```

4.2 Element `<header>`

Im `<header>`-Element werden administrative Angaben zum Dokument, zum Datenlieferanten und zu der vom Lieferanten verwendeten Software beschrieben. Die Kind-Elemente bestehen aus einer Sequenz der Elemente `<document>`, `<care_provider>` und `<software>`. Das optionale Element `<validation_status>` wird im weiteren Verlauf des Dokumentes beschrieben.

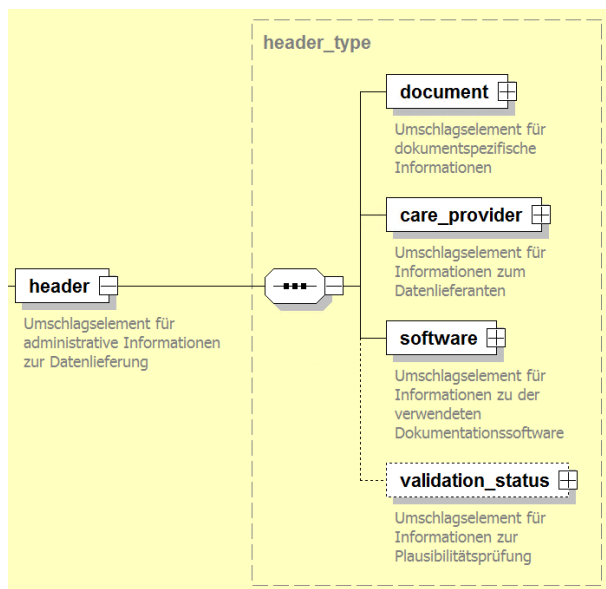


Abbildung 4: Element `<header>` mit Kind-Elementen

4.2.1 Element <header>/<document>

Das <document>-Element unter <header> bzw. deren Kind-Elemente dienen der Beschreibung bzw. eindeutigen Identifizierung des Dokuments.

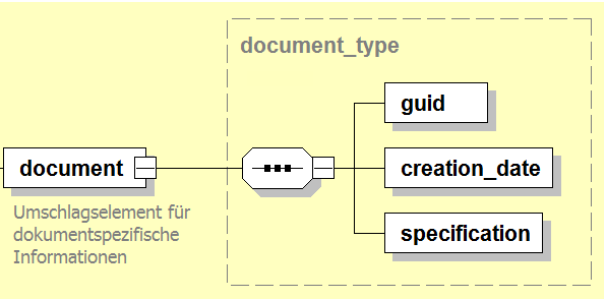


Abbildung 5: Element <document> mit Kind-Elementen

Folgende Attribute sind unter dem Element <document> verpflichtend (use = required).

Tabelle 4: Beschreibung der Kind-Elemente von <document>

| Element/Attribut | Type | Use | Beschreibung |
|----------------------|-------------|----------|---|
| guid/@value | xs:string | required | Globally Unique Identifier (GUID) Eindeutige ID des Dokuments. Die ID wird vom Dokumentersteller generiert und ist anschließend nicht mehr modifizierbar. Die ID dient vorwiegend der Kommunikation. Jede Datenlieferung einer Einrichtung muss über eine eindeutige ID verfügen. Die mehrfache Verwendung der gleichen ID ist somit ausgeschlossen. |
| creation_date/@value | xs:dateTime | required | Erstellungsdatum/-zeit des Dokuments |
| specification/@value | xs:string | required | Version der Spezifikation auf deren Basis die Datenlieferung erfolgt, bspw. „V03“ |

Der Wert im Attribut @value des Elements <guid> entspricht dem folgenden Muster/Pattern (8-4-4-4-12):

([0-9a-fA-F]){8}-([0-9a-fA-F]){4}-([0-9a-fA-F]){4}-([0-9a-fA-F]){4}-([0-9a-fA-F]){12}

Beispiel:

```
<header>
  <document>
    <guid value="2d72c378-cf41-4647-a3ad-a30243412345"/>
    <creation_date value="2024-08-15T12:00:23Z"/>
    <specification value="V03"/>
  </document>
  ... weitere administrative Daten
</header>
```

4.2.2 Element <header>/<care_provider>

Das <care-provider>-Element unter <header> bzw. deren Kind-Elemente dienen der Beschreibung bzw. eindeutigen Identifizierung des Datenlieferanten bzw. der entsprechenden Einrichtung sowie den Bezug der Datenlieferung zu einem spezifischen Zyklus.

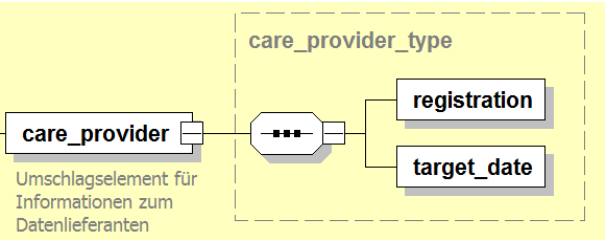


Abbildung 6: Element <care_provider> mit Kind-Elementen

Folgende Attribute sind unter dem Element <care_provider> verpflichtend (use = required).

Tabelle 5: Beschreibung der Kind-Elemente von <care_provider>

| Element/Attribut | Type | Use | Beschreibung |
|---------------------|---------|----------|--|
| registration/@value | xs:int | required | Eindeutige Einrichtungs-ID (Registrierungsnummer) die der Einrichtung während des Registrierungsprozesses bei der DAS Pflege zugewiesen wird. Die Einrichtungs-ID ist unveränderlich. Auf Grund des Datentyps sind führende Nullen ausgeschlossen. Format: 6-stellig und ganzzahlig Beispiel: 100001 |
| target_date/@value | xs:date | required | Einrichtungsindividueller Stichtag und letzter Tag des Erhebungszeitraumes der aktuellen Erhebung Format: YYYY-MM-TT |



Achtung
Mit Hilfe der Angabe im Element <target_date> erfolgt die Zuordnung einer Datenlieferung zu einem spezifischen einrichtungsindividuellen Stichtag. Sofern der angegebene Stichtag nicht mit einem bei der DAS Pflege hinterlegten Stichtag für die Pflegeeinrichtung übereinstimmt, wird die Datenlieferung abgewiesen.

Beispiel:

```
<header>
...
  <care_provider>
    <registration value="100000"/>
    <target_date value="2023-03-15"/>
  </care_provider>
...
</header>
```

4.2.3 Element <header>/<software>

Das <software>-Element unter <header> bzw. deren Kind-Elemente dienen zur Identifizierung der verwendeten Dokumentationssoftware auf Seiten der Pflegeeinrichtung.

Die Angaben zur verwendeten Software dienen vorwiegend zur Validierung des Gesamtsystems. Sofern systematische Fehler auftreten können diese anhand der Angaben ggf. auf eine spezifische Software bzw. Softwareversion zurückgeführt werden und eine Kontaktaufnahme von Seiten der DAS Pflege mit dem entsprechenden Softwarehersteller ist möglich. Hierbei sind Herstellername und Softwarename verpflichtend anzugeben.

Sofern es sich bei der Software um eine Eigenentwicklung auf Seiten der Pflegeeinrichtung handelt, kann der Einrichtungsname unter `provider/@value` eingetragen werden. Der Softwarename ist in diesem Fall von untergeordneter Relevanz und kann frei gewählt werden.

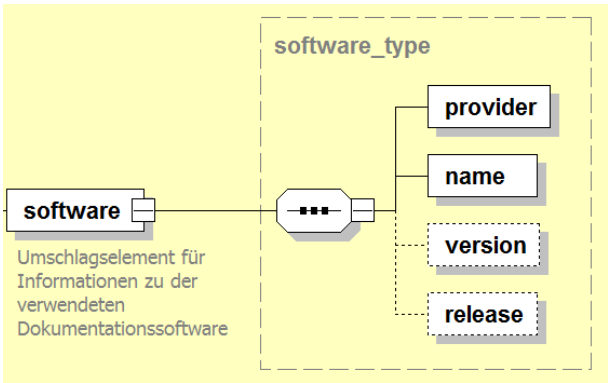


Abbildung 7: Element <software> mit Kind-Elementen

Folgende Attribute sind unter dem Element <software> zulässig.

Tabelle 6: Beschreibung der Kind-Elemente von <software>

| Element/Attribut | Type | Use | Beschreibung |
|------------------|-----------|----------|---|
| provider/@value | xs:string | required | Name des Softwareherstellers Maximale Länge: 255 Zeichen |
| name/@value | xs:string | required | Name der Software Maximale Länge: 255 Zeichen |
| version/@value | xs:string | optional | Version der Software Maximale Länge: 255 Zeichen |
| release/@value | xs:string | optional | Release der Software Maximale Länge: 255 Zeichen |

4.3 Element <body>

Das Element <body> bildet das Umschlagselement für jegliche Art von QS-Daten und die Kommentierung. Auf der ersten Ebene sind zwei Arten von Lieferungen zu unterscheiden die durch das Vorliegen des entsprechenden Kind-Elements definiert werden. Durch das Strukturelement `<xs:choice>` kann ausschließlich eines der folgend aufgeführten Elemente als Kind von <body> in einem XML-Dokument enthalten sein:

- `<data_container>` = Einrichtungs- und bewohnerbezogene QS-Daten und/oder Ausschlussgründe
- `<commentation_container>` = Kommentierung

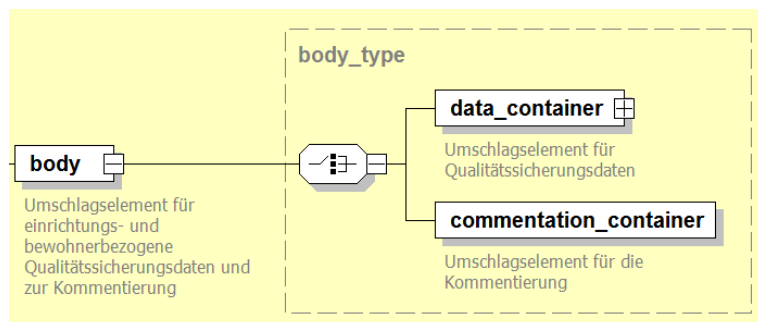


Abbildung 8: Element <body> mit Kind-Elementen

**Achtung**

Für jede Datenlieferung wird bei der Annahme geprüft, ob eine Lieferung des entsprechenden Typs zu diesem Zeitpunkt zulässig ist. Kommentierungen (definiert durch das bzw. im Element <commentation_container>) werden ausschließlich innerhalb des Kommentierungszeitraums der datenliefernden Pflegeeinrichtung angenommen. Befindet sich die Pflegeeinrichtung nicht in dieser Phase werden die Datenlieferungen abgewiesen. Gleiches gilt für Qualitätssicherungsdaten (<data_container>), die außerhalb der definierten Zeiträume übermittelt werden.

4.3.1 Element <body>/<commentation_container>

Der <commentation_container> enthält ausschließlich das Datenfeld zur Angabe des Kommentars (KOMMENTAR).

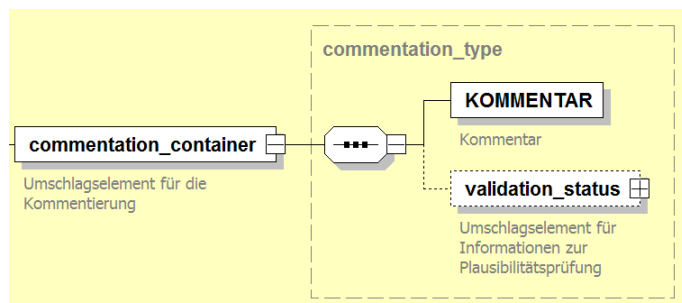


Abbildung 9: Element <commentation_container> mit Kind-Elementen

Die Kommentierung der Ergebnisse ist optional. Sofern eine Einrichtung einen Kommentar an die DAS Pflege übermittelt, wird dieser zur Veröffentlichung weitergeleitet.

**Achtung**

Kommentare müssen sachlich formuliert sein und dürfen aus Datenschutzgründen in keinem Fall personenbezogene Daten (z.B. Namen, Geburtsdatum etc.) enthalten! Zusätzlich ist zu beachten, dass der Kommentar weder von der DAS Pflege noch von den Landesverbänden der Pflegekassen vor der Veröffentlichung geprüft oder redaktionell bearbeitet wird. Ggf. kann der Anwender an geeigneter Stelle, bspw. mit der Darstellung des hinterlegten Ausfüllhinweises, auf diese Umstände explizit hingewiesen werden.

Analog zu den QS-Daten werden Kommentare mit jeder neuen Datenlieferung überschrieben. Es wird entsprechend nur der Kommentar der letzten erfolgreichen und validen Datenübermittlung verwendet.

Beispiel:

```

<body>
  <commentation_container>
    <KOMMENTAR value="Lorem ipsum dolor sit amet..."/>
  </commentation_container>
</body>

```

Sofern ein Kommentar zurückgesetzt bzw. auf Seiten der DAS Pflege gelöscht werden soll ist immer eine weitere Datenübertragung notwendig. Das Attribut @value des Elementes <KOMMENTAR> muss in diesem Fall eine leere Zeichenkette enthalten.

Beispiel:

```

<body>
  <commentation_container>
    <KOMMENTAR value=""/>
  </commentation_container>
</body>

```

4.3.2 Element <body>/<data_container>/<facility>

Die QS-Daten umfassen einrichtungsbezogene (= Daten zur Vollzähligkeitsanalyse) und bewohnerbezogene Daten. Die Unterscheidung erfolgt durch die unter <data_container> verpflichtenden Kind-Elemente <facility> und <residents> abgebildet.

- <facility> = Umschlagselement einrichtungsbezogener Daten
- <residents> = Umschlagselement bewohnerbezogener Daten

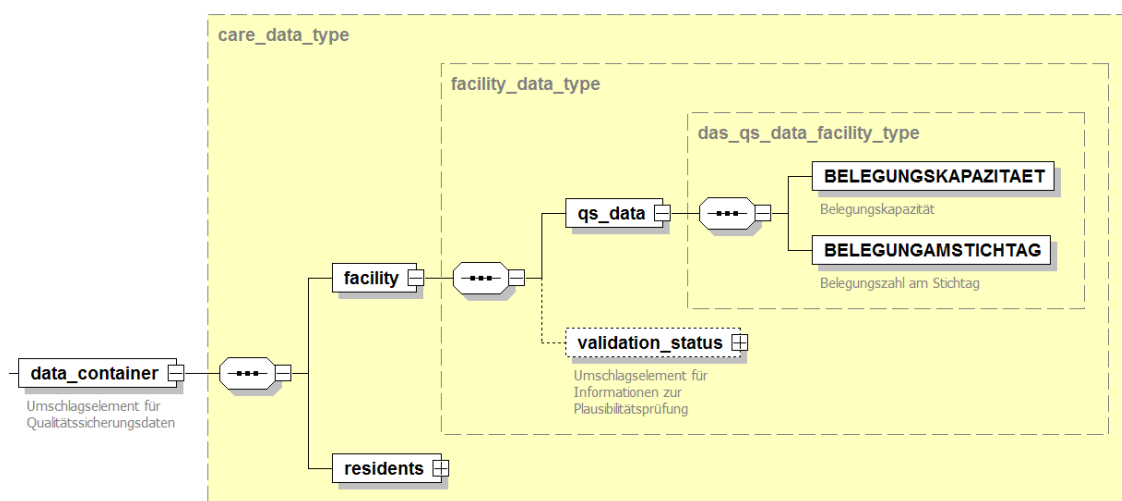


Abbildung 10: Element <facility> mit Kind-Elementen

Unter dem Element <facility> findet sich unter dem Element <qs_data> der Datensatz zur Erfassung einrichtungsbezogener Daten zur Vollzähligkeitsanalyse. Hierbei handelt es sich um verpflichtende Angaben. In der Regel sind diese Datenfelder erst nach dem Stichtag valide zu dokumentieren.

Die Definition des Datensatzes findet sich in der Schema-Datei `das_qs_data_facility.xsd`.

Beispiel:

```
<body>
  <data_container>
    <facility>
      <qs_data>
        <BELEGUNGSKAPAZITAET value="52"/>
        <BELEGUNGAMSTICHTAG value="51"/>
      </qs_data>
    </facility>
    <residents>
      ... bewohnerbezogene Daten
    </residents>
  </data_container>
</body>
```

4.3.3 Element `<body>/<data_container>/<residents>`

Das Element `<residents>` kann eine unbegrenzte Anzahl der Kind-Elemente `<resident>` enthalten. Als Kind-Elemente von `<resident>` sind wiederum ausschließlich die Elemente `<qs_data>` oder `<qs_data_mds>` zulässig. Durch die Unterscheidung wird definiert, ob es sich um eine komplette Dokumentation oder die Angabe eines Ausschlussgrundes in Bezug auf die aktuelle Erhebung handelt.

Die Definitionen der Datensätze finden sich in den Schema-Dateien `das_qs_data_type.xsd` und `das_qs_data_mds_type.xsd`.

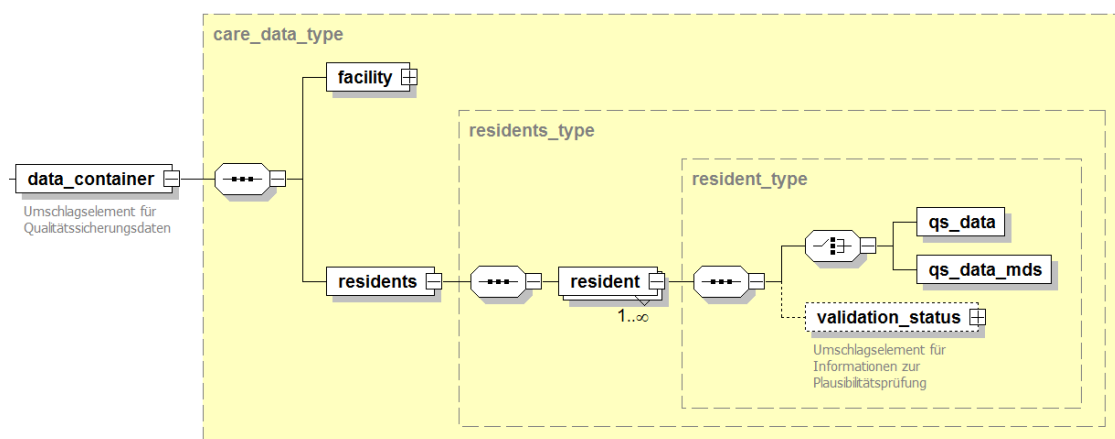


Abbildung 11: Element `<residents>` mit Kind-Elementen

Tabelle 7: Beschreibung der Kind-Elemente von `<residents>`

| Element | Use | Beschreibung |
|--------------------------|----------|--|
| <code>resident</code> | required | Das Element <code><resident></code> ist das Umschlagselement für die QS-Daten zu einem spezifischen Bewohner |
| xs:choice | | |
| <code>qs_data</code> | required | Dokumentation eines kompletten Datensatzes |
| <code>qs_data_mds</code> | required | Dokumentation eines Ausschlussgrundes zu Zeitpunkt der Erhebung |

**Achtung**

In den Kind-Elementen `<qs_data>` und `<qs_data_mds>` findet sich das Element `IDBEWOHNER`. Das Datenfeld beinhaltet die einrichtungsinterne und eindeutige bewohnerbezogene Nummer. Bezogen auf eine Datenlieferung bzw. das Element `<residents>` darf eine bewohnerbezogene Nummer nur einmal auftreten. Sollten bewohnerbezogene Nummern mehrfach enthalten sein wird die komplette Datenlieferung abgelehnt.

4.4 Element `<qs_data>`

Das Element `<qs_data>` fasst die zu dokumentierenden Datenfelder zu einer logischen Einheit in einer spezifisch festgelegten Reihenfolge zusammen. Generell weisen die Datensätze eine flache Hierarchie auf, d.h. innerhalb eines Datensatzes sind auf der zweiten Ebene keine Elemente definiert. Jeder Datensatz stellt einen komplexen Datentyp dar. Die komplette Struktur der Datensätze ist dem Schema oder dem in der Spezifikation enthaltenen Muster-Dokumentationsbogen zu entnehmen. Auf eine detaillierte Beschreibung der Datensätze bzw. der enthaltenen Datenfelder wird in der Technischen Dokumentation verzichtet.

Beispiel:

```
<body>
  <data_container>
    <facility>
      <qs_data>
        <BELEGUNGSKAPAZITAET value="52"/>
        <BELEGUNGAMSTICHTAG value="51"/>
      </qs_data>
    </facility>
    <residents>
      <resident>
        <qs_data>
          <IDBEWOHNER value="000001"/>
          <WOHNBEREICH value="3"/>
          <ERHEBUNGSDATUM value="2019-06-22"/>
          <EINZUGSDATUM value="2013-05-15"/>
          <GEBURTSMONAT value="3"/>
          <GEBURTSJAHR value="2019"/>
        </qs_data>
      </resident>
      <resident>
        <qs_data_mds>
          <IDBEWOHNER value="000002"/>
        <qs_data_mds>
      </resident>
    </residents>
  </data_container>
</body>
```

4.5 Definition von Datenfeldern

Die Definition der Datenfelder erfolgt innerhalb des Schemas immer nach dem folgenden Muster.

Beispiel:

```
<xs:element name="FELDNAME">
  <xs:annotation>
    <xs:documentation>Feldbezeichnung</xs:documentation>
    <xs:appinfo>
      ... Zusatzinformationen
    </xs:appinfo>
  </xs:annotation>
  <xs:complexType>
    <xs:attribute name="value" type="DATENTYP" use="required/optional"/>
  </xs:complexType>
</xs:element>
```

Entgegen der allgemein in englischer Sprache definierten Strukturelemente der Spezifikation werden die Elementnamen der Datenfelder auf Grund der teilweise verwendeten Fachterminologie nicht übersetzt.

Gültige Zeichen für Feldnamen (Attribut `name` im Element `xs:element`) sind A-Z und 0-9 bei einer maximalen Feldlänge von 50 Zeichen. Der Container `<xs:documentation>` enthält die menschenlesbare Bezeichnung des Datenfelds bzw. die dem Datenfeld zugrundeliegende Frage.

Generell sind Elemente, die Datenfelder abbilden nicht optional. Lediglich das Attribut `@value` kann unter bestimmten Voraussetzungen optional definiert sein. Die Elemente sind somit immer in der XML-Datei anzugeben, auch wenn das Attribut `@value` optional ist. Im anderen Fall wird die Schemavalidierung fehlschlagen.

4.5.1 Einschränkungen (Facetten)

Teilweise werden für das Attribut `@value` neue Datentypen definiert, indem einem Basis-Datentyp Einschränkungen (Facetten) hinzugefügt werden. Jede der Facetten wird als Kind-Element nach der optionalen eingebetteten `<xs:simpleType>`-Definition definiert. Damit wird bei einigen Datenfeldern der lexikalische, sowie der Werteraum eingeschränkt.

Beispiel:

```
<xs:complexType>
  <xs:attribute name="value" use="optional">
    <xs:simpleType>
      <xs:restriction base="xs:int">
        <xs:totalDigits value="2"/>
        <xs:minInclusive value="2"/>
        <xs:maxInclusive value="10"/>
      </xs:restriction>
    </xs:simpleType>
  </xs:attribute>
</xs:complexType>
```

Folgende Facetten werden derzeit verwendet.

Tabelle 8: Facetten

| Facette | Beschreibung |
|-----------------|--|
| xs:minInclusive | legt den minimalsten Wert (inklusive) fest der angegeben werden darf |
| xs:maxInclusive | legt den maximalsten Wert (inklusive) fest der angegeben werden darf |
| xs:totalDigits | legt die maximale Anzahl von Stellen für den Datentyp <code>xs:decimal</code> und davon abgeleitete Typen fest |
| xs:pattern | Definition von Mustern regulärer Ausdrücke über dem lexikalischen Raum |
| xs:maxLength | Festlegung der größten zulässigen Länge (i.d.R. für den Datentyp <code>xs:string</code>) |

Eine weitere Einschränkung von Wertebereichen auf endliche Mengen erfolgt über die Zuordnung von Facetten vom Typ `<xs:enumeration>`. Einfache Datentypen mit der Einschränkung `<xs:enumeration>` sind in der Spezifikation immer mit dem Präfix `enum_` gekennzeichnet.

Entsprechende Listen bzw. Datentypen sind der Datei `das_qs_datatypes.xsd` zu entnehmen.

Beispiel:

```
<xs:simpleType name="enum_jn_type">
  <xs:annotation>
    <xs:documentation>ja/nein</xs:documentation>
  </xs:annotation>
  <xs:restriction base="xs:int">
    <xs:enumeration value="0">
      <xs:annotation>
        <xs:documentation>nein</xs:documentation>
      </xs:annotation>
    </xs:enumeration>
    <xs:enumeration value="1">
      <xs:annotation>
        <xs:documentation>ja</xs:documentation>
      </xs:annotation>
    </xs:enumeration>
  </xs:restriction>
</xs:simpleType>
```

4.5.2 Muss/Kann-Felder (required und optional)

In Bezug auf die Datenfelder ist im Rahmen der Spezifikation zwischen Pflichtfeldern (Muss-Felder) und optionalen Kann-Feldern zu unterscheiden. Die Unterscheidung bzw. die Definition erfolgt über das Attribut `@use` des dem Element zugeordneten Attributs `@value`:

- Muss-Feld: `use="required"`
- Kann-Feld: `use="optional"`

Die Angabe bezieht sich ausschließlich auf das Attribut `@value`. Die Elemente der Datenfelddefinition sind grundsätzlich nicht optional. D.h. falls ein optionales Kann-Feld nicht gefüllt ist, ist davon unabhängig das entsprechende Element in die XML-Datei einzufügen.

4.5.3 Listenfelder

Für einige Datenfelder ist die Angabe von mehreren Werten möglich (= Listenfelder). Die maximale Anzahl der Elemente wird über das Attribut `@maxOccurs` der Elementdeklaration definiert.

Beispiel:

```
<xs:element name="DIAGNOSEN" maxOccurs="5">
...
</xs:element>
```

4.5.4 Zusatzinformationen

Neben `<xs:documentation>` wird der Container `<xs:appinfo>` für die Angabe datenfeldspezifischer und optionaler Zusatzinformationen verwendet. Die folgende Tabelle zeigt die derzeit definierten Zusatzinformationen.

Tabelle 9: Datenfeldspezifische Zusatzinformationen in `<xs:appinfo>`

| Element | Beschreibung |
|---|---|
| <code><das:ref_number></code> | Referenz zur Datenfeldnummer (Tabelle1; Spalte 1) gemäß dem dargestellten Erhebungsinstrument in Anlage 3 der Maßstäbe und Grundsätze |
| <code><das:caption_id></code> | Laufende ganzzahlige Nummer des Fragebogenabschnitts (1,2,3 etc.) |
| <code><das:caption_header></code> | Überschrift des Fragebogenabschnitts (z.B. BI-Modul 1: Mobilität) |
| <code><das:unit></code> | Angabe der Einheit bei numerischen Datenfeldern (z.B. cm, Tage etc.) |
| <code><das:guideline></code> | Ausfüllhinweis des Datenfeldes |

Zur Formatierung der Textbausteine in Element `<das:guideline>` wird rudimentäres HTML verwendet. Ein Ausfüllhinweis ist immer in ein HTML `<div>`-Tag eingeschlossen. Darüberhinausgehend werden ausschließlich folgende HTML-Tags verwendet:

- `<p>` = Textabsatz
- `` = Text Fett
- `<i>` = Text Kursiv
- `` = Ungeordnete Liste
- `` = Listenelement

Beispiel:

```
<xs:appinfo>
  <das:caption_id>1</das:caption_id>
  <das:caption_header>Allgemeine Angaben</das:caption_header>
  <das:guideline>
    <div>
      <p>Wenn das genaue Datum nicht bekannt ist,...</p>
    </div>
  </das:guideline>
</xs:appinfo>
```

4.5.5 Abhängigkeiten von Datenfeldern - Feldgruppen

Feldgruppen beschreiben logische Abhängigkeiten von Datenfeldern. Teilweise sind Angaben (= abhängige Datenfelder) nur verpflichtend und zulässig, wenn für ein anderes Datenfeld (= Filterfeld) eine oder mehrere Bedingungen zutreffen. Derartige Abhängigkeiten können beliebig tief verschachtelt sein.

Ein Beispiel für eine Feldgruppe ist die Abhängigkeit zwischen den Datenfeldern `AOPLEX` und `AOPLEXDATUM`. `AOPLEXDATUM` wird ausschließlich dann zur Pflichtangabe, wenn im Datenfeld `Apoplex` der Wert 1 = ja angegeben ist.

Die Feldgruppen-Logik wird über die Definition von Plausibilitätsregeln abgebildet und validiert.

4.6 Schemavalidierung

Die DAS Pflege führt bei jeder Datenannahme eine Schemaprüfung auf Basis der zu diesem Erhebungszyklus gültigen Spezifikation durch. Damit wird sichergestellt, dass ausschließlich inhaltlich valide Datensätze zur Auswertung herangezogen werden. Sofern die übermittelten Dateien nicht den technischen Vorgaben entsprechen oder die Validierung gegen das XML-Schema fehlschlägt, wird die gesamte Datenlieferung abgelehnt. Grundsätzlich gilt, dass im Fehlerfall immer die gesamte Datenlieferung auf Seiten der DAS Pflege nicht verarbeitet wird.

4.6.1 Regeln innerhalb des XML-Schemas (fachliche Fehler)

Innerhalb des XML-Schemas werden Plausibilitätsregeln über Assertions definiert. Die Assertions formulieren mittels XPath-Ausdrücken Bedingungen an den Inhalt von Elementen bzw. Attributen. Das Dokument ist nur dann valide, wenn alle Bedingungen erfüllt werden. Diese Regeln werden im Kontext der Spezifikation unter dem Begriff „fachliche Fehler“ zusammengefasst.

Mit dem `<assert>`-Element werden Regeln für komplexe Typen nach der folgenden Syntax formuliert: `<assert test="XPath"/>`. Der im `@test`-Attribut formulierte XPath-Ausdruck wird während der Validierung zu WAHR oder FALSCH evaluiert. Liefert er FALSCH zurück, sind die geforderten Bedingungen nicht erfüllt und das Dokument demnach nicht valide. Die in XSD 1.1 verwendete XPath-Version ist XPath 2.0.

Im Rahmen der Spezifikation ist zwischen zwei Typen von Plausibilitätsregeln zu unterscheiden:

- hart = ERROR
- weich = WARNING

Wenn mindestens eine Regel des Typs ERROR im Rahmen der Schemaprüfung verletzt wird, ist das Dokument nicht valide. Regeln des Typs WARNING sind dagegen als Nutzerhinweise auf Unstimmigkeiten in der Dokumentation zu verstehen und führen grundsätzlich nicht zur Ablehnung eines Datensatzes.

In der Spezifikation werden auch Regeln über Assertions abgebildet, die über die „Standardprüfung“ der XSD-Parser bereits abgedeckt sind. Ein Beispiel stellt die Verletzung von definierten Einschränkungen (Facetten) dar. Folgende datenfeldbezogene Fehlerarten werden zusätzlich über XPath-Ausdrücke abgefangen:

- Prüfung auf ausgefüllte Muss-Felder (`use="required"`)
- Prüfung auf Wertebereiche auf Basis von Schlüsselwerten (`xs:enumeration`)
- Prüfung auf korrekte Datentypen
- Prüfung auf Wertebereiche (`xs:minInclusive`, `xs:maxInclusive`)

Dies erfolgt mit dem Ziel, benutzerfreundliche und aussagekräftige Fehlermeldungen (vor allem in deutscher Sprache mit Bezug zu den spezifischen Datenfeldern) zu erhalten. XSD-Parser werden in diesen Fällen in der Regel Fehlermeldungen doppelt ausgeben. Zum einen die bereits über die definierten Einschränkungen resultierenden Fehlermeldungen und zum anderen die über Assertions zusätzlich abgebildeten Regeln.

Folgendes Beispiel zeigt das `<assert>`-Element einer Plausibilitätsprüfung auf eine Feldgruppenlogik.

Beispiel:

```
<xs:assert test="if(FRAKTUR/@value = 1 and not(exists(FRAKTURDATUM/@value))) then
false() else true()">
  <xs:annotation>
    <xs:appinfo>
      <das:rule_id value="60004"/>
      <das:rule_text value="[...]Spezifischer Fehlertext[...]" />
      <das:rule_type value="ERROR"/>
      <das:rule_fields>
        <das:field value="FRAKTUR"/>
        <das:field value="FRAKTURDTUM"/>
      </das:rule_fields>
    </xs:appinfo>
  </xs:annotation>
</xs:assert>
```

Plausibilitätsregeln sind immer derart definiert, dass der XPath-Ausdruck den unerwünschten Zustand, also die unplausible Angabe abbildet. Aus diesem Grund sind die Bedingungen in eine

```
if(<XPath-Bedingung>) then false() else true()
```

Anweisung eingebettet.

Für die Angabe regelspezifischer Zusatzinformationen wird analog zu den Datenfeldern der Container `<xs:appinfo>` verwendet.

Tabelle 10: Regelspezifische Zusatzinformationen in `<xs:appinfo>`

| Element | Beschreibung |
|--------------------------------------|---|
| <code><das:rule_id></code> | Interne ID der Regel |
| <code><das:rule_text></code> | Textliche Beschreibung des Fehlers bzw. der verletzten Regelbedingung |
| <code><das:rule_type></code> | Typ der Regel Folgende Ausprägungen sind möglich: „harte“ Regel = ERROR „weiche“ Regel = WARNING |
| <code><das:rule_fields></code> | Umschlagselement für die Kind-Elemente <code><das:field></code> . Sofern es das Element <code><das:rule_fields></code> gibt ist immer mindestens ein <code><das:field></code> enthalten. Hier werden die an der Regel beteiligten Datenfelder (Elemente) mit ihrem Feldnamen referenziert. Damit ist es bspw. möglich in Eingabemasken die fehlerhaften Felder hervorzuheben. |

4.6.2 Regeln außerhalb des XML-Schemas (technische Fehler)

Generell wird die Standard-Schemaprüfung des verwendeten XSD-Parsers durchgeführt. In diesem Fall ist im Kontext der Spezifikation von „technischen Fehler“ die Rede. Daneben finden sich allerdings individuelle Regeln, die zum einen nicht über das `<assert>`-Element abgebildet werden können oder die in der Standard-Schemaprüfung nicht enthalten sind. Überwiegend handelt es sich dabei um Regeln, die auf den bestehenden Datenbestand bei der DAS Pflege zurückgreifen (müssen), bspw. ob sich eine Einrichtung in einem

bestimmten Zeitraum befindet oder der entsprechende Stichtag existiert. Diese Fehler werden durch eine gesonderte Regelart repräsentiert.

4.6.3 Regelarten

In Bezug auf die Regel IDs wurde eine Kategorisierung vorgenommen, sodass aus der ID direkt ein Bezug zu einer spezifischen Regelart hergestellt werden kann. Grundsätzlich sind die Regel-IDs 5-stellig und ganzzahlig. Die folgende Tabelle zeigt die mit den Regel-IDs assoziierten Regeltypen.

Tabelle 11: Regelarten

| ID | Art der Regel | Bezug |
|-------|--|----------------------------------|
| 1xxxx | Prüfung auf ausgefüllte Muss-Felder (Pflichtfelder) | feldbezogen |
| 2xxxx | Prüfung auf gültige Schlüsselwerte (<i>xs:enumeration</i>) | feldbezogen |
| 3xxxx | Prüfung auf korrekte Datentypen | feldbezogen |
| 4xxxx | Prüfung auf Wertebereiche (Min, Max) | feldbezogen |
| 5xxxx | Prüfung von Datumsangaben | feldbezogen und feldübergreifend |
| 6xxxx | Prüfung von Feldgruppen (Abhängigkeiten) | feldübergreifend |
| 7xxxx | Sonstige Plausibilitätsprüfungen | feldbezogen und feldübergreifend |
| 9xxxx | Interne Plausibilitätsprüfungen (nicht im Schema enthalten) | feldbezogen und feldübergreifend |

4.7 Validierungsprotokoll

Die Protokollierung inhaltlicher und technischer Fehler erfolgt direkt in den gelieferten XML-Dateien, indem an der fehlerverursachenden Stelle das Element `<validation_status>` eingefügt wird. Die Antwort des Datenservice auf eine Datenlieferung besteht somit in der Regel aus dem Eingangsdokument, welches um die spezifischen Informationen der Validierung erweitert wird (= Validierungsprotokoll).

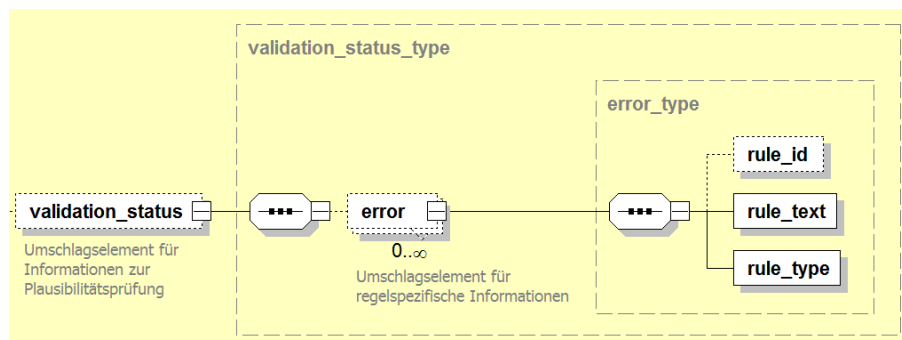


Abbildung 12: Element `<validation_status>` mit Kind-Elementen

Mögliche fehlerverursachende Zweige im XML-Dokument sind die Elemente (inkl. Kind-Elemente) `<header>`, `<commentation_container>`, `<facility_data>` und `<resident>`. Entsprechend ist das Auftreten des optionalen Elements `<validation_status>` nur an diesen Stellen möglich. Durch dieses Vorgehen können Warnungen und Fehler direkt einem Bewohnerdatensatz zugeordnet und die Information dem Anwender entsprechend aufbereitet zur Verfügung gestellt werden.

Das Element `<validation_status>` besitzt ein Attribut `@value` welches ausschließlich die folgenden Werte annehmen kann:

- WARNING = alle aufgetretenen Regelverletzungen sind vom Typ WARNING
- ERROR = mindestens eine Regelverletzung ist vom Typ ERROR

Sofern einer dieser Status zutrifft, werden die Fehlermeldungen oder Hinweise unter Nennung der `<rule_id>` (sofern definiert), des Fehlertextes (`<rule_text>`) und des Regeltyps (`<rule_type>`) unter dem Element `<validation_status>` aufgeführt.

Beispiel:

```
<validation_status value="ERROR">
  <error>
    <rule_id value="50001"/>
    <rule_text value="Regeltext"/>
    <rule_type value="ERROR"/>
  </error>
</validation_status>
```

Technische Fehler sind immer vom Typ `ERROR`. Eine Unterscheidung in die oben genannten Fehlertypen erfolgt an dieser Stelle nicht. Zusätzlich ist für diese Fehlerart keine `<rule_id>` hinterlegt. Aus diesem Grund ist die Angabe der `<rule_id>` grundsätzlich optional.

Neben dem Element `<validation_status>` ist direkt unter dem `<root>`-Element das Element `<delivery_status>` definiert. Das Element ist Bestandteil jedes Validierungsprotokolls und beinhaltet den `http-Status` der Datenlieferung und ggf. technische Fehler, die während der Validierung keiner fehlerverursachenden Quelle zugeordnet werden können.

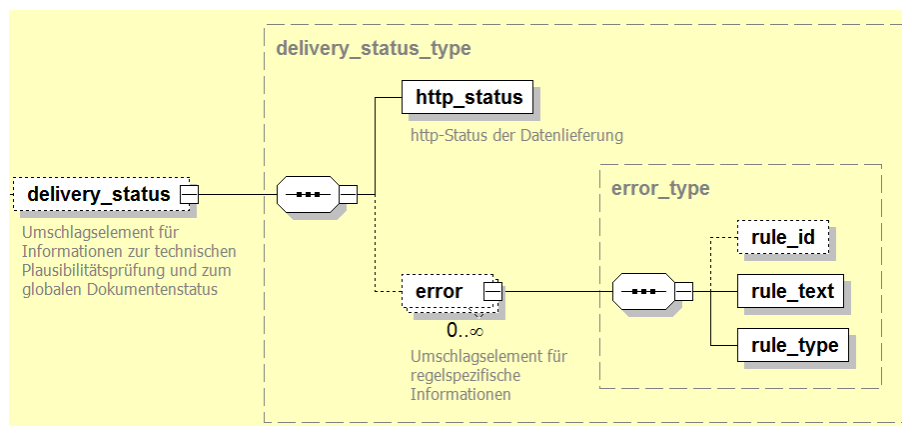


Abbildung 13: Element `<delivery_status>` mit Kind-Elementen

Im Gegensatz zum Element `<validation_status>` kennzeichnet das Attribut `@value` im Fall von `<delivery_status>` den Status der gesamten Datenlieferung. Das Attribut kann auf dieser Ebene die folgenden Ausprägungen annehmen.

- OK = Es gibt keine Regelverletzungen im gesamten Dokument; das Dokument ist valide
- WARNING = Alle Regelverletzungen sind vom Typ `WARNING`; das Dokument ist valide
- ERROR = Mindestens eine Regelverletzung ist vom Typ `ERROR`; das Dokument ist nicht valide

Aus dem Element `<delivery_status>` bzw. dem enthaltenen Attribut und dem Element `<http_status>` ist somit der globale Status des Dokumentes zu entnehmen. Im Fall der Ausprägung `ERROR` werden die Datensätze auf Seiten der DAS Pflege nicht persistiert, d.h. an dieser Stelle sind zwingend Korrekturen an der Datei vorzunehmen und diese erneut an den Datenservice zu übermitteln.

**Achtung**

Der Eintrag des Status OK oder WARNING im Element `<delivery_status>` in Kombination mit dem Eintrag 200 im Element `<http_status>` ist somit eine Bestätigung der Validität der Datenlieferung und der Verarbeitung auf Seiten des Datenservice bzw. der DAS Pflege.

Ist ein Eingangsdokument nicht validierbar (z.B. bei falschem Datentyp) liefert der Datenservice ein XML-Dokument mit folgendem Aufbau (sog. Miniprotokoll):

Beispiel:

```
<root>
  <delivery_status value="ERROR">
    <http_status value="400"/>
    <error>
      <rule_text value="Fehlermeldung"/>
      <rule_type value="ERROR"/>
    </error>
  </delivery_status>
</root>
```

5 Tabellarische Darstellung der Zeiträume und Fristen

Die tabellarische Darstellung der Zeiträume und Fristen im CSV-Format ermöglicht es Anwendern alle relevanten Daten bzw. Datumsangaben in Abhängigkeit vom gewählten Stichtag einer Einrichtung zu entnehmen. Die Darstellung erfolgt bundeslandbezogen, da für die Berechnung landesspezifische Feiertage (z.B. der Reformationstag) berücksichtigt werden. Sukzessive wird die Liste um weitere Zyklen erweitert. Die folgende Tabelle beschreibt die im Tabellenformat enthaltenen Spalten.

Tabelle 12: CSV-Format - Zeiträume und Fristen

| Spalte | Beschreibung |
|---|--|
| Bundesland | Bundesland der Einrichtung; die Unterscheidung nach Bundesland ist notwendig auf Grund länderspezifischer Feiertage |
| Stichtag1 | Erster gewählter Stichtag (ohne Jahresbezug) |
| Stichtag2 | Zweiter gewählter Stichtag (ohne Jahresbezug) |
| Erhebungszyklus | Aufsteigende laufende Nummer des Erhebungszyklus beginnend mit dem 1. Halbjahr 2021 (= 1). <ul style="list-style-type: none"> Jeder Stichtag ist eindeutig einem Erhebungszyklus zugeordnet. Jedem Erhebungszyklus ist eine Spezifikationsversion zugeordnet, wobei eine Version über mehrere Zyklen bestand haben kann. |
| XML-Path: root.header.care_provider.target_date | Stichtagsdatum welches bei Datenlieferungen im Element <target_date> einzutragen ist |
| Erhebung Beginn | Beginn des Erhebungszeitraums |
| Erhebung Ende | Ende des Erhebungszeitraums |
| Ergebniserfassung Beginn | Beginn des Ergebniserfassungszeitraums |
| Ergebniserfassung Ende | Ende des Ergebniserfassungszeitraums |
| Auswertung1 Beginn | Beginn der Auswertungsphase 1 |
| Auswertung1 Ende | Ende der Auswertungsphase 1 |
| Korrektur Beginn | Beginn des Korrekturzeitraums |
| Korrektur Ende | Ende des Korrekturzeitraums |
| Auswertung2 Beginn | Beginn der Auswertungsphase 2 |
| Auswertung2 Ende | Ende der Auswertungsphase 2 |
| Kommentierung Beginn | Beginn des Kommentierungszeitraums |
| Kommentierung Ende | Ende des Kommentierungszeitraums |

6 Webservice (REST-API)

Die Datengrundlage des Portals DAS Pflege stellt die zentrale Datenbank dar, in der die erfassten Bewohnerdatensätze (QS-Daten) und die produzierten Auswertungen (Berichte oder Reports) verwaltet werden. Das Portal der DAS Pflege unterstützt mehrere Kommunikationskanäle für das Erfassen und das Lesen der Daten. Die folgende Matrix stellt die Nutzer (Akteure) in Relation dazu dar, ob es sich um einen Personen- oder Maschinenzugang handelt.

Tabelle 13: Matrix - Personen- vs. Maschinenzugang

| Akteur | Datenerfassung | | | Datenkonsum | | |
|-------------------|----------------|--------|-----------------|-------------|--------|-----------------|
| | Portal | Upload | Webservice | Portal | Upload | Webservice |
| Pflegeeinrichtung | Person | Person | Maschine | Person | - | Maschine |
| Landesverband | - | - | - | Person | - | Maschine |
| Prüfdienst | - | - | - | Person | - | Maschine |

Die DAS Pflege stellt Dritten eine Möglichkeit zur Verfügung, über Softwareschnittstellen (= Maschinenzugang) mit dem System zu kommunizieren. Zentrale Funktionalitäten der Schnittstelle sind die Datenerfassung von Bewohnerdaten sowie der Abruf von Berichten im XML-, CSV- und PDF-Format.

Es handelt sich um eine REST-Schnittstelle über HTTP(S)³. Zum Datenaustausch wird dabei auf JSON⁴ und XML/XSDs gesetzt.

6.1 Client Implementierungen

Softwareanbieter sind in der Gestaltung ihrer Softwareprodukte frei, haben allerdings bei der Umsetzung zwingend die Vorgaben der Spezifikation (beispielsweise Datensatzbeschreibung, Datenprüfung und -Übermittlung) zu beachten.

Auch bei der Wahl der Technologien und der Programmiersprachen zur Anbindung der REST-API sind die Softwareanbieter frei. Nahezu alle gängigen Sprachen/Tools unterstützen die Möglichkeit, HTTP-Aufrufe an entfernte Server zu schicken und viele Tools bieten auch die Möglichkeit, JSON in einfacher Art zu verarbeiten.

6.2 Testumgebung

Für Softwareanbieter steht eine Testumgebung für die REST-API zur Verfügung. Die Softwareversion auf der Testumgebung entspricht dem Stand der produktiven Umgebung www.das-pflege.de. Die Testumgebung beinhaltet feste Testdaten gegen die Softwareanbieter Ihre Implementierungen testen können.

Der Host für die Testumgebung ist: api.das-pflege.de

³ Hypertext Transfer Protocol (Secure)

⁴ <https://www.json.org/>

6.3 REST-API-Dokumentation

Die REST-API-Dokumentation beschreibt technische Details für die Nutzung der REST-API in drei wesentlichen Aspekten:

1. Accept-Header & Content-Type-Header: Abhängig des Endpunkts werden Anfragen und Antworten im in verschiedenen Formaten (i.d.R. XML oder JSON, für Berichte auch PDF und Text) erstellt/ausgegeben.
2. Die notwendigen Schritte zum Authentisieren und anschließendem Autorisieren
3. Die Endpunkte mit Angabe der Parameter, Rückgabewerte und Responses und der Möglichkeit, sich cURL-Aufrufe generieren zu lassen und Test-Requests an den jeweiligen Endpunkt zu schicken.

Die Dokumentation ist unter folgender Adresse erreichbar:

<https://api.das-pflege.de/api/doc/index.html>