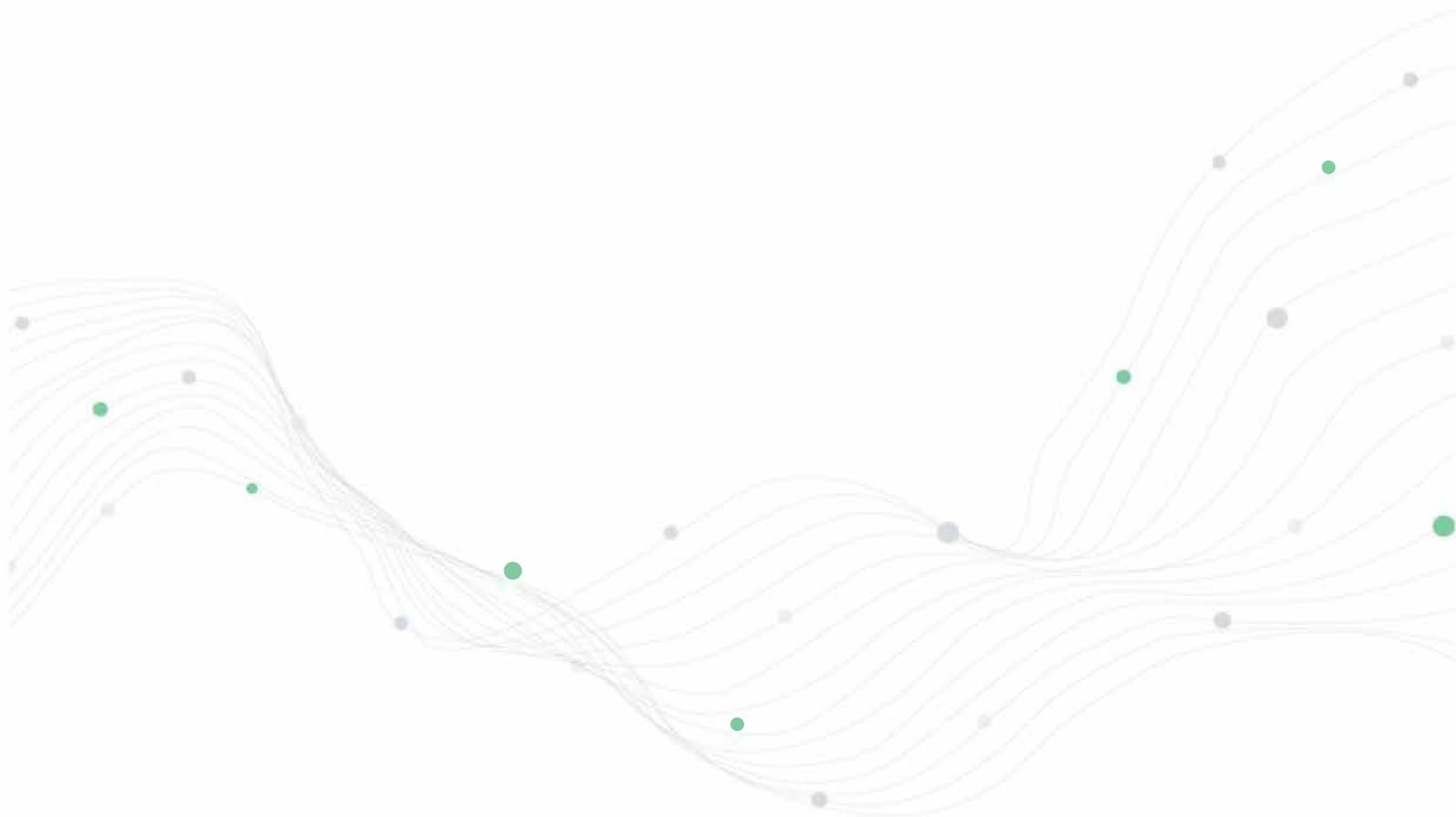


# ISBJ TRÄGERPORTAL

ENTWICKLERLEITFADEN FÜR DIE DIENSTSCHNITTSTELLE VERSION  
1.19.0

DOKUMENTVERSION 1.15.0



# Versionshistorie

*Tabelle 1:* Versionshistorie

Datum	Änderungen	Dokumentenversion		Autor
07.07.22	<ul style="list-style-type: none"> <li>• zusammenführen von "v7. 74.0 <i>Entwicklerleitfaden für die Dienstschnittstelle Version 1.18.0 Personalplanung</i>" &amp; "v13.0 <i>/SBJ Trägerportal Entwicklerleitfaden Dienstschnittstelle 1.13.0</i>"</li> <li>• Korrektur der Dokumente</li> <li>• Redesign</li> <li>• Hinzufügen Validierung Backend Personalplanung</li> <li>• Aktualisierung der REST-Schnittstellen Doku</li> <li>• Überprüfung der Richtigkeit</li> <li>• Kapitel "Referenzclient" entfernt, da dies Teil der Testphase war</li> </ul>	1.15.0	Final	Jakob Schäfer

# Inhaltsverzeichnis

<b>1</b>	<b>Einleitung</b>	<b>5</b>
<b>2</b>	<b>Grundlagen</b>	<b>6</b>
2.1	Fachliche Grundlagen	6
2.2	Organisatorische Grundlagen	6
2.2.1	Beteiligte Personen und Rollen	6
2.2.1.1	Betreiber des ISBJ Trägerportals und der zugehörigen Dienstschnittstelle	6
2.2.1.2	Mandanten des ISBJ Trägerportals und der zugehörigen Dienstschnittstelle	7
2.3	Technische Grundlagen	7
2.4	Daten und Informationen	8
<b>3</b>	<b>Umgebung des ISBJ Trägerportals und der zugehörigen Dienstschnittstelle</b>	<b>9</b>
3.1	Umgebungen	9
3.1.1	Abnahmeumgebung	9
3.1.2	Produktivumgebung	9
3.2	Zugang zum ISBJ Trägerportal	9
3.2.1	Daten für die Betreiber des ISBJ Trägerportals	10
3.2.2	Checkliste zur Kommunikation mit der Dienstschnittstelle	10
3.3	Aufruf des Trägerportals im Webbrowser	11
3.4	Anmeldung am Trägerportal als Nutzer der Dienstschnittstelle	11
3.5	REST-Aufrufe an die Dienstschnittstelle	12
3.5.1	Autorisierung	12
3.5.2	Smoketest Beispiel	13
<b>4</b>	<b>Entwicklung eines Clients</b>	<b>15</b>
4.1	Datenlieferung und Abfrage	15
4.1.1	Datenbestand	15
4.1.2	Status- und Detailabfragen	15
4.1.3	Asynchrone Datenlieferung	16
4.1.4	Technische Validierung	16
4.1.5	Fachliche Validierung und Protokollabfragen	16
4.1.6	Prüfsummenvalidierung und -berechnung	16
4.1.6.1	Wichtige Informationen	16
4.1.6.2	Technische Definition	17
4.1.6.3	Berechnung der Prüfsumme	17
4.1.6.4	Prüfsumme für Personalplanung	20
4.2	REST-Schnittstelle	22
4.2.1	XSD-Schema	22
4.2.2	URL	22
4.2.3	HTTP-Request-Header	22

4.2.4	HTTP-Request-Body	23
4.2.4.1	XML-Header	23
4.2.4.2	XML-Body	23
4.2.5	Anwendungsfälle und Funktionen	25
4.2.5.1	REST-WebServices	26
4.2.5.2	XSD-Schema technische Validierung bei Datenlieferung	31
4.2.5.3	Identifikation von Person bei personalmeldung	32
4.2.5.4	Abfrage von Statusinformationen einzelner Datensätze nach einer Lieferung	33
4.2.6	Response-Codes	34
4.2.7	Eigenschaften der Dienstschnittstelle	34
4.2.7.1	Datenlieferung	34
4.2.7.2	Reaktionszeiten auf verschiedene Aktionen bei Lieferungen	35
4.2.7.3	Zusätzliche Informationen zum XSD-Schema	35
4.2.7.4	Fehlermeldungen und deren Weitergabe	36

## Abbildungsverzeichnis

1	Trägerportal Anmeldung .....	11
2	Portal Anmeldung .....	11
3	Erfolgreiche Anmeldung .....	12
4	Benutzerdetails .....	12
5	Smoketest cURL Befehl .....	13
6	Smoketest Anfrage .....	13
7	Smoketest Antwort .....	14
8	Prüfsumme Beispiel 1 (real) .....	18
9	Prüfsumme Beispiel 2 (frei gewählt) .....	20
10	URL-Aufbau in BNF .....	22
11	XML-Header Beispiel .....	23

## Tabellenverzeichnis

1	Versionshistorie .....	1
2	Relevante Datenhaltungsorte für Statusabfragen je nach Anwendungsfall .....	15
3	Anwendungsfälle smoketest & protoko/1 .....	26
4	Anwendungsfall freiplatzmeldung .....	26
5	Anwendungsfall kindzuschlag .....	27
6	Anwendungsfall kitaverzeichnis .....	27
7	Anwendungsfall personalplanung .....	28
8	Anwendungsfall stellenangebot .....	29
9	Anwendungsfall vertrag .....	30
10	Anwendungsfall vormerkung .....	30
11	Aktionen für die Funktion lieferung .....	31
12	Validierungen personalplanung .....	32
13	Status einzelner Datensätze .....	33
14	Status einer Lieferung .....	33
15	Response-Codes .....	34
16	Reaktionsverhalten auf Aktionen bei Lieferungen .....	35

# 1 Einleitung

Dieses Dokument richtet sich an Entwickler, die eine bestehende Fachanwendung über die Dienstschnittstelle an das ISBJ Trägerportal anbinden möchten. Es bietet eine detaillierte Anleitung zur Vorbereitung einer Clientanwendung für die Kommunikation mit der Dienstschnittstelle. Die Dienstschnittstelle bietet Zugriff auf bestimmte Funktionen des ISBJ Trägerportals, die von der Clientanwendung aufgerufen werden können. Das Aufrufen der Dienstschnittstelle über das Internet erfolgt mittels REST-Service (REST: Representational State Transfer), welcher als Transportprotokoll HTTP(S) verwendet. Das Nachrichtenformat ist vom Typ XML. Die Form wird durch eine Schemadefinitionsdatei (XSD) vorgegeben.

Der vorliegende Leitfaden befasst sich sowohl mit organisatorischen Grundlagen als auch mit den technischen Aspekten der Anbindung. Er beschreibt, wie die Dienstschnittstelle programmatisch angesprochen wird, welche Daten in welchem Format vorliegen müssen und welche Rückmeldungen zu erwarten sind.

## 2 Grundlagen

### 2.1 Fachliche Grundlagen

Die erfolgreiche Anbindung einer bestehenden Fachanwendung an die Dienstschnittstelle erfordert detaillierte Kenntnisse der Fachanwendung und der ISBJ-Anwendungslandschaft.

Besteht über die Fachanwendung oder über die ISBJ-Anwendungslandschaft Informationsbedarf, kann über den Träger-Service Kontakt mit der Senatsverwaltung für Bildung, Jugend und Familie (SenBJF) aufgenommen werden. So können frühzeitig fachliche Fragen beantwortet bzw. relevante fachliche Konzepte oder Auszüge daraus zur Verfügung gestellt werden. Eine Implementierung auf rein technischer Ebene ist zwar möglich, dieses Vorgehen kann jedoch zu unnötigen mehrfachen Implementierungsversuchen führen.

### 2.2 Organisatorische Grundlagen

Da es sich beim Trägerportal und dessen Dienstschnittstelle als Teil des ISBJ-Fachverfahrens um ein komplexes System mit verschiedenen Rollen und Rechten handelt, ist es notwendig, die für die Nutzung der Dienstschnittstelle relevanten Rollen abzugrenzen. Des Weiteren ist es erforderlich, alle relevanten Technologien aufzuzählen, welche bei der Kommunikation mit der Dienstschnittstelle in Verwendung sind. Dies dient dazu, einen Rahmen zu errichten, in welchem die Entwicklung einer externen Clientanwendung stattfinden kann.

#### 2.2.1 Beteiligte Personen und Rollen

Für die Anbindung von Clientanwendungen an die Dienstschnittstelle ist die Kommunikation zwischen den Mandanten und den Betreibern des ISBJ Trägerportals und dessen Dienstschnittstelle ein integraler Bestandteil des Prozesses. Daher ist es wichtig, die Verantwortlichkeiten auf beiden Seiten klar zu definieren. Die unten beschriebenen Rollen müssen nicht auf unterschiedliche Personen verteilt sein.

##### 2.2.1.1 Betreiber des ISBJ Trägerportals und der zugehörigen Dienstschnittstelle

Der **Mandanten-Betreuer (ISBJ-Träger-Service)** ist die Person, die direkt für den Austausch von Daten und Dateien zwischen Betreibern und Mandanten zuständig ist. Konkret wird diese Aufgabe durch den Träger-Service von SenBJF erfüllt. Der Träger-Service hat die Aufgabe, Mandanten-Informationen entgegenzunehmen und in der Portal-Benutzerverwaltung (Portal-BNV) zu pflegen. Es geht dabei insbesondere um das Erstellen eines Administrators und das Ausstellen des zugehörigen Zertifikats. Gleichzeitig liefert der Träger-Service Informationen, die die Mandanten für die Anbindung benötigen. Dazu gehören die Zugangsdaten des Administrators sowie das Zertifikat.

Weiterhin ist der Betreiber dafür verantwortlich, dass die Firewall mit den Zertifikatsinformationen gepflegt wird, um die Kommunikation zu ermöglichen.

### 2.2.1.2 Mandanten des ISBJ Trägerportals und der zugehörigen Dienstschnittstelle

Der **Prozessverantwortliche (Administrator im Trägerportal)** ist eine administrative Rolle im Betrieb des Mandanten, der in Absprache mit dem Mandanten-Betreuer die Nutzerrechte und Zugänge für den jeweiligen Träger definiert.

Dieser Person werden das Zertifikat und die Zugangsdaten für den Administrator im Trägerportal zugewiesen. Anschließend ist der Prozessverantwortliche zuständig für die Einrichtung von Benutzern und von deren Rechten. Es muss für die Clientanwendung mindestens ein Benutzer im Trägerportal eingerichtet sein, da ohne diesen Benutzer nicht mit der Clientanwendung auf die Dienstschnittstelle zugegriffen werden kann.

**Dienstschnittstellen-Benutzer** haben die Möglichkeit, unterschiedliche Anwendungsfälle über die Dienstschnittstelle auszuführen.

**Entwickler** sind die Personen in der Organisation des Mandanten, die mit der Erstellung und Pflege der Clientanwendung betraut sind. Sie sind die primäre Zielgruppe dieses Dokuments, da sie für die programmatische Anbindung der Clientanwendung verantwortlich sind. Zu ihren Aufgabenbereichen zählt auch die Einbindung der Zertifikate.

## 2.3 Technische Grundlagen

Es werden keine Programmierkenntnisse in einer bestimmten Sprache für die Verwendung der Dienstschnittstelle vorausgesetzt, da die Kommunikation über eine REST-Schnittstelle erfolgt. Grundlegende Kenntnisse in Java sind jedoch empfehlenswert, wenn versucht wird, die Kommunikation mithilfe des Referenzclients und anhand von Programmierbeispielen nachzuvollziehen. Ggf. sind Maven-Kenntnisse erforderlich, wenn der Referenzclient gebaut werden soll.

Vorausgesetzt werden allgemeine Kenntnisse in den folgenden Bereichen:

- **REST** - Grundlegende Kenntnisse des Konzepts und der verschiedenen Methodenarten sollten vorhanden sein.
- **TLS** - Diese Verschlüsselungsmethoden werden zur Sicherung der Übertragung verwendet. Es genügen Kenntnisse im Umgang mit den Verschlüsselungsmethoden.
- **XML** - Grundlegende XML-Kenntnisse sind erforderlich.
- **XSD** - In XSD genügt es, das vorgegebene Schema und v. a. darin festgelegte Formate und Restriktionen zu verstehen. Für den Fall, dass in einer Sprache Modellklassen aus dem XSD generiert

werden sollen, sind ggf. Kenntnisse im Umgang mit entsprechenden Bibliotheken für die jeweilige Sprache nötig. Bei Java sind beispielsweise Erfahrungen im Umgang mit JAXB hilfreich.

- **Fachliches Wissen** - Um ungültige Anfragen aufgrund fachlich falscher Daten zu vermeiden, empfiehlt es sich, diese bereits vor dem Versand abzufangen. Anderenfalls werden die fachlich ungültigen Daten mit entsprechenden Fehlercodes, welche später noch beschrieben werden, quittiert.

## 2.4 Daten und Informationen

Folgende Informationen sind für die Verwendung des Trägerportals über die Dienstschnittstelle erforderlich:

- **Benutzerzugang** - Er dient zum Einspielen und Auslesen von Daten über die Dienstschnittstelle. Er besteht aus Nutzernamen, Kennwort und zugehörigem Zertifikat.
- **Zertifikat** - Es dient zur Authentifizierung des Nutzers.
- **URL des Trägerportals** - Über die URL können die Benutzer der Dienstschnittstelle die eigenen Rechte einsehen.
- **URL des REST-Services** - Über diese URL können die verschiedenen Services der Dienstschnittstelle aufgerufen werden.
- **Übersicht der Services** - Sie beinhaltet eine Aufzählung, welche Dienste von der Dienstschnittstelle bereitgestellt werden.
- **XSD-Schema** - Es beschreibt die Form der XML-Dateien, die die Dienstschnittstelle entgegennimmt oder die als Antwort auf Abfragen generiert werden.

Woher diese Informationen stammen und wie sie im Prozess der Dienstschnittstellennutzung verwendet werden, wird in den folgenden Kapiteln vermittelt.

## 3 Umgebung des ISBJ Trägerportals und der zugehörigen Dienstschnittstelle

### 3.1 Umgebungen

- **Abnahmeumgebung** - Die Zugangsinformationen zur Abnahmeumgebung müssen separat mit dem Träger-Service abgestimmt werden. Diesbezüglich kann eine E-Mail mit dem Betreff "Dienstschnittstelle" an [traeger-service@senbjf.berlin.de](mailto:traeger-service@senbjf.berlin.de) gesendet werden.
- **Produktivumgebung**- Der Beantragungsprozess für die Zugangsdaten des Produktivsystems muss zwischen dem Dienstleister und dessen Auftraggeber abgestimmt werden.

#### 3.1.1 Abnahmeumgebung

Die Abnahmeumgebung wird im IT-Dienstleistungszentrum Berlin (ITDZ) betrieben. Für die Vergabe der Benutzer und Zertifikate ist der ISBJ-Träger-Service zuständig. Diese Umgebung ist zu verwenden, um echte HTTPS-Zugriffe mit Clientzertifikaten zu testen und die Korrektheit der Clientanwendung gegen eine abzunehmende Version der Dienstschnittstelle zu verifizieren. Die hierfür nötigen Zertifikate und Benutzerkennungen können bei SenBJF beantragt werden.

Positive Tests auf der Abnahmeumgebung sind die Voraussetzung für die Produktivumgebung des Clients.

#### 3.1.2 Produktivumgebung

Auch die Produktivumgebung wird im ITDZ betrieben. Hier sollten die Träger bereits Zertifikate und Administratorkennungen beantragt haben, damit weitere Benutzer über das Trägerportal eingerichtet werden können. Für neue Administratorkennungen muss die Rolle Dienstschnittstelle vom Träger-Service SenBJF zugeordnet sein, um sie Benutzern vererben zu können.

### 3.2 Zugang zum ISBJ Trägerportal

In diesem Abschnitt wird beschrieben, welche Schritte zur Verwendung der Dienstschnittstelle notwendig sind. Dabei wird an den entsprechenden Stellen auf die Unterschiede zwischen den zwei Umgebungen hingewiesen. Die ersten Schritte dienen dazu zu prüfen, ob von den jeweiligen Rechnern bzw. vom Netzwerk auf das Trägerportal zugegriffen werden kann. Es wird ebenso aufgezeigt, wie ein Benutzer anzulegen ist bzw. wie die Rechte eines Benutzers zu überprüfen sind, bevor versucht wird, mit diesem auf die Dienstschnittstelle zuzugreifen.

### 3.2.1 Daten für die Betreiber des ISBJ Trägerportals

Zuerst müssen die Nutzerdaten für die jeweilige Umgebung angefordert werden. Diese Informationen werden per E-Mail an den Betreiber übermittelt. Zu übermitteln sind jeweils:

- Nachname
- Vorname
- E-Mailadresse
- Telefon
- Anschrift (Straße, Hausnummer, PLZ, Ort)
- Name und/oder Nummer des Trägers, für den der Zugang eingerichtet werden soll
- Name und/oder Nummern der Einrichtungen, für die der Zugang eingerichtet werden soll
- Auflistung der Rechte, welche der Benutzer für die Einrichtungen benötigt

### 3.2.2 Checkliste zur Kommunikation mit der Dienstschnittstelle

Bevor mit der Dienstschnittstelle gearbeitet werden kann, sollte sichergestellt sein, dass alle relevanten Zugangsdaten für die Nutzung der Dienstschnittstelle vorhanden sind. Hier ist eine Checkliste aller Voraussetzungen:

- Haben Sie die Nutzerdaten vom Betreiber erhalten?
  - Liegt Ihnen Ihr Nutzernamen vor?
  - Liegt Ihnen Ihr Nutzerkennwort vor?
- Haben Sie ein Zertifikat vom Betreiber zugewiesen bekommen?
  - Liegt Ihnen das Zertifikat in PKCS12-Format (dateiname.p12) vor?
  - Liegt Ihnen das Installationskennwort des Zertifikats als Schreiben oder PDF-Dokument vor?
- Haben Sie die aktuelle Version der XML-Schemadefinition (isbj-dienstschnittstelle-1.19.0.xsd) der Dienstschnittstelle erhalten?
- Haben Sie die URL für die Zugriffe erhalten?
  - URL des ISBJ Trägerportals
  - URL der Dienstschnittstelle

Falls eine oder mehrere Voraussetzungen fehlen, muss der Betreiber kontaktiert werden.

### 3.3 Aufruf des Trägerportals im Webbrowser

Nach Erhalt der benötigten Zugangsdaten kann das Trägerportal unter der übermittelten URL aufgerufen werden.

Hierbei ist darauf zu achten, dass das übermittelte Zertifikat im verwendeten Browser installiert ist. Unterstützung bei der Installation des Zertifikats kann über den Träger-Service von SenBJF [traeger-service@senbjf.berlin.de](mailto:traeger-service@senbjf.berlin.de) erbeten werden.

### 3.4 Anmeldung am Trägerportal als Nutzer der Dienstschnittstelle

Das Portal empfängt Sie mit folgender Startseite:

Abbildung 1-Trägerportal Anmeldung



Klicken Sie rechts oben auf Anmelden. Sie werden zu der Portal-Anmeldung weitergeleitet:

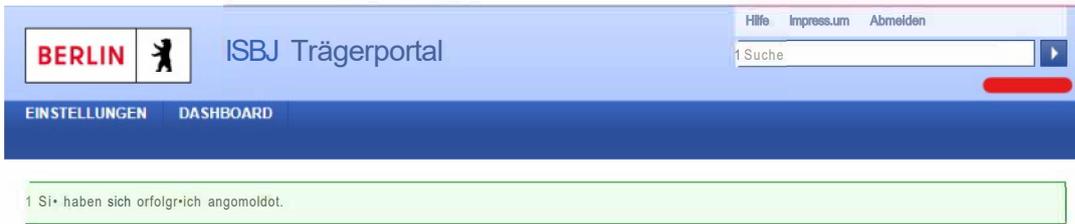
Abbildung 2- Portal Anmeldung



Tragen Sie in der folgenden Eingabemaske den übermittelten Benutzernamen und das zugehörige Passwort ein.

Nachdem Sie sich erfolgreich angemeldet haben, wird die Anmeldung bestätigt und der Menüpunkt Einstellungen freigeschaltet.

Abbildung 3 Erfolgreiche Anmeldung



Durch das Klicken auf Einstellungen wird das Untermenü Eigene Details sichtbar.

Die Auswahl des Menüpunkts Eigene Details führt zu einer Übersicht, welche die übermittelten Nutzerdaten enthält und die Rechte des Nutzers nach Trägern auflistet.

Abbildung 4: Benutzerdetails



Prüfen Sie an dieser Stelle, ob die Stammdaten richtig übernommen wurden, die Rolle Dienstschnittstelle zugewiesen ist und ob alle erforderlichen Rechte vergeben wurden.

Sind alle Daten korrekt, ist Ihr Benutzer für die Dienstschnittstelle eingerichtet.

## 3.5 REST-Aufrufe an die Dienstschnittstelle

An dieser Stelle wird ein einfacher REST-Aufruf für die Dienstschnittstelle am Anwendungsfall smoketest skizziert.

### 3.5.1 Autorisierung

Der Verbindungsaufbau zur Dienstschnittstelle findet über eine verschlüsselte TLS-Verbindung mittels HTTPS statt. Auf Antrag eines Datenlieferanten vergibt ein Administrator ein Clientzertifikat aus der Zertifikatsverwaltung. Der Antragsteller bekommt das Zertifikat (PKCS12-Format) per E-Mail zugestellt. Das Installationskennwort bekommt der Antragsteller aus Sicherheitsgründen per Post.

Das Zertifikat muss für den TLS-Verbindungsaufbau mit dem Dienstschnittstelle-Server verwendet werden.

Nachdem ein Datenlieferant mit einem gültigen Clientzertifikat die Verbindung aufbaut, muss er seine Nutzerkennung angeben, bevor Aktionen in der Dienstschnittstelle durchgeführt werden können. Im Trägerportal werden die Nutzerkennung in der Oberfläche eingegeben, in der REST-Schnittstelle wird die Nutzerkennung per HTTP-Header-Einträge übermittelt (BASIC-Authentification nach [RFC 7677](#)).

### 3.5.2 Smoketest Beispiel

Für den Aufruf sollten jetzt folgende Daten bereitliegen:

- Benutzername
- Benutzerpasswort
- Zertifikatsdatei in PKCS72-Format
- Zertifikat-Passwort
- URL der Dienstschnittstelle
- Anwendungsname (Standard: porta/-ws)

Die Anfrage-URL ergibt sich wie folgt

```
<protoko/1>://<URL>:<Port>/<anwendungsname>/rest/smoketest
```

Ein Beispiel für eine Anfrage und Antwort in cURL inklusive Clientzertifikat sieht wie folgt aus:

Abbildung 5: Smoketest cURL Befehl

```
#1 /bin/bash
# cert-file: cert.p12
# cert-password: password
# user: dss
# user-password: password
# url: https://dss-url.de/portal-ws/rest/smoketest
curl --cert-type P12 --cert cert.p12:password \
--location --request GET 'https://dss-url.de:8080/portal-ws/rest/smoketest' \
--header 'Authorization: Basic ZHNzOnBhc3N3b3Jk'
```

Abbildung 6: Smoketest Anfrage

```
GET /portal-ws/rest/smoketest HTTP/1 .1
Host: https://dss-url.de:8080
Accept: */*
Authorization: Basic ZHNzOnBhc3N3b3Jk
```

*Abbildung 7* Smoketest Antwort

```
HTTP/1.1 200 OK
Server: Apache-Coyote/1.1
Content-Type: application/xml;charset=UTF-8
Content-Length: 760
Date: Wed, 22 Jun 2022 16:17:48 GMT

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<root xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation="isbj-dienstschnittstelle-1.19.0.xsd">
  <header xsi:type="header-protokoll_type">
    <erstellungsdatum>2022-06-22T14:15:14</erstellungsdatum>
    <Software>
      <hersteller-name>Schütze Consulting AG</hersteller-name>
      <software-name>ISBJ Dienstschnittstelle</software-name>
      <software-version>1.19.0</software-version>
    </Software>
    <protokoll>
      <status>OK</status>
    </protokoll>
  </header>
  <body xsi:type="body-smoketest_type">
    <smoketest-antwort>
      <smoketest-aktion>Anmeldung am Traegerportal</smoketest-aktion>
      <status>OK</status>
      <smoketest-meldung>Vorname: XXX Nachname: XXX</smoketest-meldung>
    </smoketest-antwort>
  </body>
</root>
```

## 4 Entwicklung eines Clients

Sind die oben aufgeführten organisatorischen und technischen Grundlagen erfüllt und alle Zugangsdaten für die Nutzung der Dienstschnittstelle bekannt, kann mithilfe der folgenden Informationen die Gegenseite zur Dienstschnittstelle implementiert werden.

### 4.1 Datenlieferung und Abfrage

Im Folgenden wird das Verhalten der Dienstschnittstelle in Bezug auf Datenlieferungen und Abfragen beschrieben. Des Weiteren werden Details über die verwendeten Datenformate erläutert, um eine korrekte Verwendung der Dienstschnittstelle zu ermöglichen.

#### 4.1.1 Datenbestand

Der Datenbestand des Trägerportals kann unter bestimmten Bedingungen nicht immer aktuell sein. Das hängt mit dem Abgleich der Daten aus den Backend-Anwendungen zusammen, welcher die Daten aus dem gesicherten Netzwerk mit den Daten aus dem Trägerportal abgleicht. Normalerweise sind die Daten des Trägerportals aufgrund dieses regelmäßigen Datenabgleichs nicht älter als 24 Stunden, die Daten der endgültigen Datenhaltung wiederum nicht älter als 48 Stunden.

#### 4.1.2 Status- und Detailabfragen

Abfragen auf Daten werden unterteilt in Statusabfragen und Detailabfragen. Detailabfragen beinhalten sämtliche fachliche Daten, wie sie auch im Trägerportal einsehbar sind. Statusabfragen liefern hingegen Metainformationen wie beispielsweise die URL und das Datum einer Veröffentlichung. Statusabfragen liefern erst dann ein Ergebnis bzw. das aktuelle Ergebnis zurück, wenn die Daten in der endgültigen Datenhaltung vorhanden sind. Diese unterscheiden sich je nach Anwendungsfall:

*Tabelle 2:* Relevante Datenhaltungsorte für Statusabfragen je nach Anwendungsfall

Anwendungsfall für Statusabfrage	Ort der endgültigen Datenhaltung	Replikation
personalplanung	ISBJ E&D Personal	innerhalb 24h
vormerkliste	ISBJ Vormerkliste	innerhalb 24h
stel lenangebot	berlin.de	innerhalb 48h
freiplatzmeldung	berlin.de	innerhalb 48h
kitaverzeichnis	berlin.de	innerhalb 48h

Details zu den verschiedenen Status sowie Detailantworten können der Schemadefinition unter `antwort_type` entnommen werden.

Für alle Abfragen an die Dienstschnittstelle gilt Es können nur Daten abgefragt werden, für die der abfragende Nutzer die Rechte hat.

### **4.1.3 Asynchrone Datenlieferung**

Datenlieferungen werden asynchron bearbeitet, d. h., die Lieferung wird angenommen und mit einer Vorgangsnummer (auch Trackingnummer genannt) bestätigt, sofern die XML-Daten dem Schema entsprechen. Erst danach werden diese Daten validiert und ggf. an das Trägerportal weitergeleitet, von wo aus diese zu den Backend-Anwendungen weitergegeben werden. Das Ergebnis der Validierung steht wenige Minuten nach der Lieferung über den Aufruf des Anwendungsfallsprotokoll zur Verfügung. Durch die Abfrage der Protokollinformationen einer Lieferung mithilfe der Trackingnummer kann die Validität der einzelnen Datensätze und die der gesamten Datenlieferung nachvollzogen werden.

Fachliche Antworten bzw. Statusaktualisierungen finden im Rahmen der normalen Replikationen von Backend-Anwendungen zum Trägerportal statt.

### **4.1.4 Technische Validierung**

Vor jeder fachlichen Validierung wird eine technische Validierung durch die Dienstschnittstelle durchgeführt. Dabei werden z.B. die Schemakonformität, UTF-8-Kodierung der Daten, Korrektheit der Nutzerdaten und Gültigkeit des Nutzerzertifikats geprüft.

Sollte die technische Validierung fehlschlagen, wird die gesamte Lieferung abgelehnt. Eine entsprechende Fehlermeldung wird als Antwort auf den Request gesendet.

### **4.1.5 fachliche Validierung und Protokollabfragen**

Das Trägerportal führt eine Reihe von fachlichen Validierungen durch, die analog zu den Eingabevalidierungen über die Oberfläche des Trägerportals die fachliche Korrektheit von Datensätzen verifizieren. Sind die Daten valide, werden sie unverändert an die Backend-Anwendungen weitergeleitet.

Für den Fall, dass Datensätze in einer Lieferung fachlich nicht valide sind, werden diese Datensätze nicht bearbeitet. Entsprechende Fehlermeldungen und Status einzelner Datensätze können aus den Protokollabfragen zu Datenlieferungen entnommen werden.

### **4.1.6 Prüfsummenvalidierung und -berechnung**

#### **4.1.6.1 Wichtige Informationen**

Die Prüfsummenberechnung kann konfigurierbar ein- und ausgeschaltet werden. Dies liegt im Ermessen des Betreibers des ISBJ Trägerportals. Bitte kontaktieren Sie hierzu den Betreiber.

**Sonderfall:** Die Prüfsummenvalidierung und -berechnung ist ein Feature, das für den Anwendungsfall *personalplanung* immer deaktiviert ist (seit Version 1.18.0)<sup>1</sup>. In der Lieferung können Dummywerte eingefügt werden, diese müssen weiterhin dem XSD-Schema entsprechen (siehe Kapitel 4.1.6.2). Ein Beispiel ist unter Kapitel 4.1.6.4 zu finden.

#### 4.1.6.2 Technische Definition

Eine Prüfsumme ist ein durch das MD5-Hash Verfahren verschlüsselter String.

Eine Prüfsumme ist 32 Zeichen lang und besteht aus der Menge der Hexadezimal-Zahlen mit lower-case Buchstaben (Regex: "[a-f0-9]{32}"). Voranstehende Nullen werden aufgefüllt. Die XSD-Schema-Typ-Bezeichnung ist `pruefsumme_type`.

#### 4.1.6.3 Berechnung der Prüfsumme

Jede Datenlieferung enthält zwei Arten von MD5-Prüfsummen: jeweils eine Prüfsumme pro Datensatz und eine Prüfsumme für die gesamte Datenlieferung.

Im XML-Header steht die Prüfsumme der gesamten Datenlieferung. Sie wird über die konkatenierten Prüfsummen der einzelnen Datensätze in der Reihenfolge erstellt, in der sie im XML-Code auftreten.

Im XML-Body befinden sich die Prüfsummen für jeden Datensatz. Um sie zu bilden, werden die Einrichtungsnummer, die Empfänger-ID sowie die Inhalte aus den Fachdaten-Tags konkateniert und mit dem MD5-Hashverfahren verschlüsselt. Bei der Konkatenierung ist die Reihenfolge des Auftretens innerhalb des XML-Codes (von oben nach unten) einzuhalten.

Durch die Prüfsummenvalidierung wird die Datenintegrität bei der Übermittlung verifiziert. Des Weiteren werden diese Prüfsummen verwendet, um Dubletten zu erkennen, so dass diese abgelehnt werden. Hierzu werden:

- die Prüfsumme der Datenlieferung gegen vorhandene Datenlieferungsprüfsummen in der Datenbank überprüft. Existiert dieselbe Datenlieferungsprüfsumme wird die gesamte Datenlieferung abgelehnt.
- die Prüfsummen anhand der gelieferten Daten neu berechnet. Unterscheiden sich die gelieferten Prüfsummen von den berechneten, wird die gesamte Datenlieferung als ungültig abgelehnt.

Sind beide Prüfsummen identisch, wird in der Datenbank überprüft, ob bereits ein Datensatz mit derselben Prüfsumme existiert. In diesem Fall wird der Datensatz mit dem Fehler Dublette erkannt abgelehnt. Andernfalls folgt die fachliche Validierung.

Im folgenden Beispiel wird der XML-Code einer Lieferung an die *freiplatzmeldung* mit den dazugehörigen Prüfsummen dargestellt und anschließend die Prüfsummenberechnung erläutert:

Abbildung 8: Prüfsumme Beispiel 7 (real)

```

<root xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <header xsi:type="header-anfrage_type">
    <erstellungsdatum>2022-12-06T10:52:40</erstellungsdatum>
    <Software>
      <hersteller-name>Hersteller</hersteller-name>
      <software-name>DSS</software-name>
      <software-version>1</software-version>
    </Software>
    <pruefsumme>92cb834cd10ff39f3fdb2ec605582fe4</pruefsumme>
  </header>
  <body xsi:type="body_type">
    <traeger nummer="0001">
      <einrichtung nummer="01020050">
        <datensatz lfdnummer="1" xsi:type="datensatz-anfrage_type">
          <admin-anfrage>
            <erstellerid>0001</erstellerid>
            <erstellungsdatum>2022-12-06T10:52:40</erstellungsdatum>
            <aenderungsdatum>2022-12-06T10:52:40</aenderungsdatum>
            <aktion>create</aktion>
            <pruefsumme>80538184ae2d0a0a86a4a07017e6b74b</pruefsumme>
          </admin-anfrage>
          <fachdaten>
            <freiplatzmeldung>
              <bemerkung>Nur für Fingerfarbenprofis</bemerkung>
              <webadresse>www.colorful-fingertipps.de</webadresse>
              <gueltig-ab>2021-08-07</gueltig-ab>
              <freie-plaetze-gesamt>9</freie-plaetze-gesamt>
              <freie-plaetze-ohne-altersangabe>4</freie-plaetze-ohne-aJ
                ....ltersangabe>
              <freie-plaetze-unter-drei>2</freie-plaetze-unter-drei>
              <freie-plaetze-ueber-drei>3</freie-plaetze-ueber-drei>
              <aufnahmealter-monate>12</aufnahmealter-monate>
              <betreuungsumfang>HTB</betreuungsumfang>
            </freiplatzmeldung>
          </fachdaten>
        </datensatz>
        <datensatz lfdnummer="2" xsi:type="datensatz-anfrage_type">
          <admin-anfrage>
            <erstellerid>0002</erstellerid>

```

```

    <empfaengerid>5552</empfaengerid>
    <erstellungsdatum>2022-12-06T10:52:40</erstellungsdatum>
    <aenderungsdatum>2022-12-06T10:52:40</aenderungsdatum>
    <aktion>update</aktion>
    <pruefsumme>d4e329a15c3ad76f9d0ade3289edc90e</pruefsumme>
  </admin-anfrage>
  <fachdaten>
    <freiplatzmeldung>
      <bemerkung>Nur für Fingerfarbenprofis</bemerkung>
      <webadresse>www.colorful-fingertipps.de</webadresse>
      <gueltig-ab>2014-09-01</gueltig-ab>
      <freie-plaetze-gesamt>2</freie-plaetze-gesamt>
      <freie-plaetze-ohne-altersangabe>3</freie-plaetze-ohne-aJ
      ↵ ltersangabe>
      <freie-plaetze-unter-drei>2</freie-plaetze-unter-drei>
      <freie-plaetze-ueber-drei>3</freie-plaetze-ueber-drei>
      <aufnahmealter-monate>12</aufnahmealter-monate>
      <betreuungsumfang>GTB</betreuungsumfang>
    </freiplatzmeldung>
  </fachdaten>
</datensatz>
<datensatz lfdnummer="3" xsi:type="datensatz-anfrage_type">
  <admin-anfrage>
    <erstellerid>0003</erstellerid>
    <empfaengerid>5553</empfaengerid>
    <erstellungsdatum>2022-12-06T10:52:40</erstellungsdatum>
    <aenderungsdatum>2022-12-06T10:52:40</aenderungsdatum>
    <aktion>delete</aktion>
    <pruefsumme>58edb2ed5bfbclade2fc5cd6eaa895ab</pruefsumme>
  </admin-anfrage>
  <fachdaten />
</datensatz>
</einrichtung>
</traeger>
</body>
</root>

```

Dieses Beispiel enthält drei Datensätze vom Anwendungsfall Freiplatzmeldung mit den Aktionen create, update und delete. Darin gibt es vier Prüfsummen, welche folgendermaßen gebildet werden.

Die Prüfsummen der einzelnen Datensätze im XML-Body ergeben sich aus den konkatenierten Werten der Einrichtungsnummer, der Empfänger-ID sowie der Fachdaten des jeweiligen Datensatzes, wobei es

im create-Fall keine Empfänger-ID und im delete-Fall keine Fachdaten gibt:

- Prüfsumme **1. Datensatz** mit Aktion **create**: 80538184ae2d0a0a86a4a07017e6b74b = MD5("01020050Nur für Fingerfarbenprofiswww.colorful-fingertipps.de2021-08-01942312HTB")
- Prüfsumme **2. Datensatz** mit Aktion **update**: d4e329a15c3ad76f9d0ade3289edc90e = MD5("010200505552Nur für Fingerfarbenprofiswww.colorful-fingertipps.de2014-09-01232312GTB")
- Prüfsumme **3. Datensatz** mit Aktion **delete**: 58edb2ed5bfbcb1ade2fc5cd6eaa895ab = MD5("010200 50 5553")

Die Prüfsumme der gesamten Datenlieferung steht im XML-Header und ergibt sich aus dem MD5-Hash der Prüfsummen der einzelnen Datensätze in der Reihenfolge, wie sie im XML-Code auftreten:

- Prüfsumme **XML-Header**: 92cb834cd10ff39f3fdb2ec605582fe4 = MD5("80538184ae2d0a0a86a4a07017e6b74b" + "d4e329a15c3ad76f9d0ade3289edc90e" + "58ed b2ed5bfbcb1 ade2fc5cd6eaa89 5ab")

#### 4.1.6.4 Prüfsumme für Personalplanung

Für den Anwendungsfall personalplanung wird die Prüfsumme nicht validiert. Bei der Datenlieferung sind die Prüfsummen frei wählbar gemäß dem XSD-Schema (siehe Kapitel 4.1.6.2).

Im folgenden Beispiel wird der XML-Code einer Lieferung an die personalplanung mit frei wählbaren Prüfsummen dargestellt:

*Abbildung 9* Prüfsumme Beispiel 2 (freigewählt)

```
<root xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <header xsi:type="header-anfrage_type">
    <erstellungsdatum>2022-12-06T10:52:40</erstellungsdatum>
    <Software>
      <hersteller-name>Hersteller</hersteller-name>
      <software-name>DSS</software-name>
      <software-version>1</software-version>
    </Software>
    <pruefsumme>000000000000000000000000f0001000a</pruefsumme>
  </header>
  <body xsi:type="body_type">
    <traeger nummer="0001">
      <einrichtung nummer="01020050">
        <datensatz lfdnummer="1" xsi:type="datensatz-anfrage_type">
          <admin-anfrage>
```



## 4.2 REST-Schnittstelle

Im Folgenden wird die REST-Schnittstelle der Dienstschnittstelle genauer beschrieben.

### 4.2.1 XSD-Schema

Das XSD-Schema ist ein fundamentaler Bestandteil der REST-Schnittstelle. Es beschreibt die XML-Datentypen für Request- & Response-Bodies. Gelieferte Daten werden zunächst gegen das XSD-Schema validiert, bevor ihr Inhalt weiter fachlich geprüft wird. Im Fall invalider Daten wird zusätzlich zum Response-Code auch mit einer beschreibenden Fehlermeldung geantwortet. Um unnötigen Traffic zu vermeiden, empfiehlt es sich, die zu liefernden Daten bereits vor der Lieferung zu validieren.

Für die aktuelle Version 1.19.0 der Dienstschnittstelle wird das XSD-Schema `isbj-dienstschnittstelle-1.19.0.xsd` verwendet.

### 4.2.2 URL

Abbildung 10: URL-Aufbau inBNF

```

<url>          ::= <basis url>/<anwendungsfall>[/<funktion>][?<parameterliste>]
<basis url>    ::= <protokoll>://<host>:<port>/<anwendungsname>/rest
<protokoll>    ::= http[s]
<funktion>     ::= lieferung | abfragequote | abfragedetail
                | 1 abfrageeinrichtungohnepersonal
<parameterliste> ::= <parameter>[&<parameterliste>]
  
```

Die Verwendung der Schnittstelle erfolgt stets über eine URL in der in Abbildung 10 angegebenen Form.

Protokoll, Host, Port und Anwendungsname für die Basis-URL können beim Betreiber der Dienstschnittstelle angefragt werden. Die Anwendungsfälle mit ihren möglichen Funktionen und Parametern sind dem Abschnitt REST-Webservices zu entnehmen.

Beispiel:

`https://dss-url.de/porta1-ws/rest/personalplanung/abfragedetails?traeger=1234&einrichtung=12345678`

### 4.2.3 HTTP-Request-Header

Bei jedem Request an die Dienstschnittstelle muss die Authentifizierung als HTTP BASIC-Authentification im Request-Header übertragen werden, die dieselbe Funktion wie die Anmeldemaske im Trägerportal erfüllt. Der Wert ist eine Base64 gehashte Zeichenkette, bestehend aus Benutzernamen und Passwort, die mit einem Doppelpunkt verknüpft sind (BASIC-Authentification nach [RFC 7617](#)).

Fehlt die BASIC-Authentification im Header oder schlägt die Authentifizierung fehl, wird eine HTTP-Response mit HTTP-Response-Code 401 und einer Fehlermeldung zurückgeliefert.

Bei Datenlieferungen (mit der Funktion `lieferung`) wird zusätzlich der Custom-Header **Content-Type** erwartet, der mit dem Wert `application/xml` gesetzt ist.

Unter Kapitel "3.5.1 Autorisierung" und Kapitel "3.5.2 Smoketest Beispiel" ist die Autorisierung anhand eines Beispiels erläutert.

#### 4.2.4 HTTP-Request-Body

Ein Request-Body muss nur bei Datenlieferungen (mit der Funktion `lieferung`) übermittelt werden. Er enthält XML-Code, dessen Struktur über das Schema definiert ist.

Der XML-Code im Request-Body unterteilt sich in zwei Abschnitte, den XML-Header und den XML-Body. Im folgenden Beispiel handelt es sich um einen Datensatz des Anwendungsfalls Personalplanung.

##### 4.2.4.1 XML-Header

Im XML-Header werden die Informationen über das Erstellungsdatum, die verwendete Software und die Gesamtprüfsumme über die Prüfsummen der einzelnen Fachdatensätze übermittelt (siehe Kapitel 4.1.6 Prüfsummenvalidierung und -berechnung). Genauere Informationen können im Schema unter `header_abstract_type` und den abgeleiteten Typen entnommen werden.

*Abbildung 11: XML-Header Beispiel*

```
cheader xsi:type="header-anfrage_type">
  cerstellungsdatum>2022-12-06T10:52:40c/erstellungsdatum>
  csoftware>
    chersteller-name>Herstellerc/hersteller-name>
    csoftware-name>DSSc/software-name>
    csoftware-version>1</software-version>
  c/software>
  cpruefsumme>00000000000000000000000000000000f0001000ac/pruefsumme>
c/header>
```

##### 4.2.4.2 XML-Body

Der XML-Body besitzt die Struktur `traeger-> einrichtung -> datensatz`. Pro Lieferung ist ein Träger mit maximal 200 Einrichtung sowie jeweils 1000 Datensätzen pro Einrichtung erlaubt. Genauere Informationen können dem Schema unter `body_type` und den abgeleiteten Typen entnommen werden.

Ein Datensatz besteht jeweils aus der `admin-anfrage` und den `fachdaten`. Die Schema-Definition ist `datensatz-anfrage_type`.

Die admin-anfrage ist für alle Anwendungsfälle identisch (**admin-anfrage\_type**). Sie enthält folgende Informationen:

- **erstellerid** - ID des Fremdsystems, welche dem Fremdsystemersteller ermöglicht, die von ihm erstellten Datensätze in den ISBJ-Anwendungen wiederzufinden
  - Die Ersteller-ID gehört dem Fremdsystem und wird in die ISBJ-Anwendungen nur übernommen, um dem Fremdsystem bei Detail- und Statusabfragen die Zuordnung der Datensätze zu vereinfachen.
  - Die Prüfung der Ersteller-ID soll nur die Speicherung dieses Feldes in den ISBJ-Anwendungen garantieren und verlangt daher:
    - Die Ersteller-ID muss bei Lieferungen vorhanden sein.
    - Die Ersteller-ID muss die Form des **id\_type** haben (ganze Zahl zwischen 1 und 9999999999)
  - Das Anlegen von zwei unterschiedlichen Datensätzen mit derselben Ersteller-ID ist erlaubt.
- **empfaengerid** - ID des Zieldatensatzes, unter der die Daten in der Backend-Anwendung eindeutig hinterlegt sind
  - Die Empfänger-ID wird bei einer Datenlieferung bei der Aktion create generiert und kann über die Funktion `abfragedetails` oder `abfragestatus` abgerufen werden.
  - Der Datensatz lässt sich über die vergebene Ersteller-ID identifizieren.
  - Für die eindeutige Vergabe der Ersteller-ID ist der Nutzer der Dienstschnittstelle verantwortlich.
- **erstellungdatum** - Datum der Erstellung des Datensatzes
- **aenderungdatum** - Datum der letzten Änderung des Datensatzes
- **aktion** - Aktion zur Übermittlung der Fachdaten (create, update, delete)
- **pruefsumme** - Prüfsumme über den Inhalt der Fachdaten

Die fachdaten unterscheiden sich je nach Anwendungsfall. Informationen und Definitionen hierzu können im XSD-Schema unter **fachdaten\_type** nachgeschlagen werden. Informationen zu technischen Validierungen der fachdaten, welche nicht Bestandteil des XSD-Schemas ist, sind unter Kapitel 4.2.5.2 zu finden.

Ein Beispiel einer kompletten Datenlieferung ist in Abbildung 8 und Abbildung 9 unter Kapitel "4.7.6.3 Berechnung der Prüfsumme" zu finden.

#### 4.2.5 Anwendungsfälle und Funktionen

Für einen Anwendungsfall kann es bis zu vier verschiedene Funktionen geben. Die Funktion *lieferung* ermöglicht das Einpflegen von Daten mittels einer HTTP-POST-Methode und einem XML-Body mit den einzupflegenden Daten auf Basis des XSD-Schemas. Die Funktionen *abfragestatus* und *abfragedetail* ermöglichen das Abrufen von Informationen über die HTTP-GET-Methode. Wie im Abschnitt Status- und Detailabfragen 4.1.2 beschrieben, können mit diesen Funktionen Metainformationen oder Fachdaten abgefragt werden.

Im Einzelnen werden die Anwendungsfälle und Funktionen in Kapitel 4.2.5.1 beschrieben.

#### 4.2.5.1 REST-WebServices

Im Folgendend werden die Anwendungsfälle genauer beschrieben (Funktion, HTTP-Methode, URL-Parameter, Beschreibung). Falls URL-Parameter für einen Anwendungsfall oder eine Funktion aufgelistet ist, wird ebenfalls die Parameter-Validierung anhand von Referenzen zu XSD-Schema-simpleType hinterlegt. Für die Funktion /lieferung wird auf Referenzen zu XSD-Schema-complexType hingewiesen, welche für die fachdaten (fachdaten\_type) benötigt werden.

*Tabelle 3: Anwendungsfälle smoketest & protokoll*

Anwendungsfall	URL	Methode	Parameter	Beschreibung
smoketest	–	GET	–	Der Smoketest meldet zurück, ob die Dienste aktiv sind und validiert die Benutzerrechte.
protokoll	–	GET	<ul style="list-style-type: none"> <li>trackingnr</li> </ul>	Status und eventuelle Validierungsmeldungen zu einer Datenlieferung werden zurückgeliefert. Bei einer Lieferung mit sehr großen Datenmengen kann die Protokollabfrage das Ergebnis erst zurückgeben, nachdem alle Daten vollständig bearbeitet wurden.

*Tabelle 4: Anwendungsfall freiplatzmeldung*

Anwendungsfall freiplatzmeldung			
Funktion	Methode	Parameter	Beschreibung
lieferung	POST	–	Die Datenlieferung im Request-Body wird hochgeladen.
abfragestatus	GET	<ul style="list-style-type: none"> <li>traeger</li> <li>einrichtung (optional)</li> </ul>	Die Abfrage liefert den Status von Freiplatzmeldungen für einen Träger bzw. eine Einrichtung eines Trägers zurück. Der Parameter einrichtung ist optional.

Parameter Validation	
traeger	traegernr_type
einrichtung	einrichtungnr_type

*Fortsetzung auf der nächsten Seite*

*Tabelle 4: Anwendungsfall freiplatzmeldung (Fortsetzung)*

Anwendungsfall freiplatzmeldung									
Funktion	Methode	Parameter	Beschreibung						
abfragedetail	GET	<ul style="list-style-type: none"> <li>• traeger</li> <li>• einrichtung</li> </ul>	<p>Die Freiplatzmeldungsdaten zur abgefragten Einrichtung werden zurückgeliefert.</p> <table border="1"> <thead> <tr> <th colspan="2">Parameter Validation</th> </tr> </thead> <tbody> <tr> <td>traeger</td> <td>traegernr_type</td> </tr> <tr> <td>ein richtung</td> <td>einrichtungnr_type</td> </tr> </tbody> </table>	Parameter Validation		traeger	traegernr_type	ein richtung	einrichtungnr_type
Parameter Validation									
traeger	traegernr_type								
ein richtung	einrichtungnr_type								

*Tabelle 5: Anwendungsfall kindzuschlag*

Anwendungsfall: kindzuschlag													
Funktion	Methode	Parameter	Beschreibung										
abfragedetail	GET	<ul style="list-style-type: none"> <li>• traeger</li> <li>• einrichtung (optional)</li> <li>• rechtsgrundlage (optional)</li> <li>• kartennr (optional)</li> </ul>	<p>liefert BuT-Kindzuschläge zur abgefragten Einrichtung zurück. Die Ergebnismenge kann mit den optionalen Parameter einrichtung, rechtsgrundlage und kartennr (Berlin Passnummer) weiter eingeschränkt werden.</p> <table border="1"> <thead> <tr> <th colspan="2">Parameter Validation</th> </tr> </thead> <tbody> <tr> <td>traeger</td> <td>traegernr_type</td> </tr> <tr> <td>einrichtung</td> <td>einrichtungnr_type</td> </tr> <tr> <td>rechtsgrundlage</td> <td>rechtsgrundlage_type_enum</td> </tr> <tr> <td>kartennr</td> <td>kartennummer_type</td> </tr> </tbody> </table>	Parameter Validation		traeger	traegernr_type	einrichtung	einrichtungnr_type	rechtsgrundlage	rechtsgrundlage_type_enum	kartennr	kartennummer_type
Parameter Validation													
traeger	traegernr_type												
einrichtung	einrichtungnr_type												
rechtsgrundlage	rechtsgrundlage_type_enum												
kartennr	kartennummer_type												

*Tabelle 6: Anwendungsfall kitaverzeichnis*

Anwendungsfall: kitaverzeichnis			
Funktion	Methode	Parameter	Beschreibung
lieferung	POST	–	Die Datenlieferung im Request-Body wird hochgeladen.

*Fortsetzung auf der nächsten Seite*

*Tabelle 6: Anwendungsfall kitaverzeichnis {Fortsetzung}*

Anwendungsfall: <i>kitaverzeichnis</i>											
Funktion	Methode	Parameter	Beschreibung								
abfragestatus	GET	<ul style="list-style-type: none"> <li>• traeger</li> <li>• einrichtung (optional)</li> </ul>	<p>Die Abfrage liefert den Status von Kitaverzeichniseinträgen für einen Träger bzw. eine Einrichtung eines Trägers zurück. Der Parameter einrichtung ist optional.</p> <table border="1"> <thead> <tr> <th colspan="2">Parameter Validation</th> </tr> </thead> <tbody> <tr> <td>traeger</td> <td><b>traegernr_type</b></td> </tr> <tr> <td>ein richtung</td> <td><b>einrichtungnr_type</b></td> </tr> </tbody> </table>	Parameter Validation		traeger	<b>traegernr_type</b>	ein richtung	<b>einrichtungnr_type</b>		
Parameter Validation											
traeger	<b>traegernr_type</b>										
ein richtung	<b>einrichtungnr_type</b>										
abfragedetail	GET	<ul style="list-style-type: none"> <li>• traeger</li> <li>• einrichtung</li> <li>• bild (optional)</li> </ul>	<p>Die Kitaverzeichnisdaten zur abgefragten Einrichtung werden zurückgeliefert. Der Parameter bild ist optional und erlaubt die Werte true und false. true liefert die Daten mit Bild, sofern eines vorhanden ist. Default = false</p> <table border="1"> <thead> <tr> <th colspan="2">Parameter Validation</th> </tr> </thead> <tbody> <tr> <td>traeger</td> <td><b>traegernr_type</b></td> </tr> <tr> <td>ein richtung</td> <td><b>einrichtungnr_type</b></td> </tr> <tr> <td>bild</td> <td><b>xs:boolean</b></td> </tr> </tbody> </table>	Parameter Validation		traeger	<b>traegernr_type</b>	ein richtung	<b>einrichtungnr_type</b>	bild	<b>xs:boolean</b>
Parameter Validation											
traeger	<b>traegernr_type</b>										
ein richtung	<b>einrichtungnr_type</b>										
bild	<b>xs:boolean</b>										

*Tabelle 7: Anwendungsfall personalplanung*

Anwendungsfall: <i>personalplanung</i>									
Funktion	Methode	Parameter	Beschreibung						
lieferung	POST	–	Die Datenlieferung im Request-Body wird hochgeladen.						
abfragedetai 1	GET	<ul style="list-style-type: none"> <li>• traeger</li> <li>• einrichtung</li> </ul>	<p>Eine Liste der Personalmeldungen für die abgefragte Einrichtung wird zu rückgeliefert.</p> <table border="1"> <thead> <tr> <th colspan="2">Parameter Validation</th> </tr> </thead> <tbody> <tr> <td>traeger</td> <td><b>traegemr_type</b></td> </tr> <tr> <td>einrichtung</td> <td><b>einrichtungnr_type</b></td> </tr> </tbody> </table>	Parameter Validation		traeger	<b>traegemr_type</b>	einrichtung	<b>einrichtungnr_type</b>
Parameter Validation									
traeger	<b>traegemr_type</b>								
einrichtung	<b>einrichtungnr_type</b>								

*Fortsetzung auf der nächsten Seite*

*Tabelle 7 Anwendungsfall personalplanung (Fortsetzung)*

Anwendungsfall: <i>personalplanung</i>									
Funktion	Methode	Parameter	Beschreibung						
abfragequote	GET	<ul style="list-style-type: none"> <li>• traeger</li> <li>• einrichtung</li> </ul>	<p>Eine Liste der Personalquoten und der Personalsituation der abgefragten Einrichtung wird zurückgeliefert.</p> <table border="1"> <thead> <tr> <th colspan="2">Parameter Validation</th> </tr> </thead> <tbody> <tr> <td>traeger</td> <td><b>traegemr_type</b></td> </tr> <tr> <td>einrichtung</td> <td><b>einrichtungnr_type</b></td> </tr> </tbody> </table>	Parameter Validation		traeger	<b>traegemr_type</b>	einrichtung	<b>einrichtungnr_type</b>
Parameter Validation									
traeger	<b>traegemr_type</b>								
einrichtung	<b>einrichtungnr_type</b>								
abfrageeinrichtungunzureichendepersonal	GET	<ul style="list-style-type: none"> <li>• traeger</li> </ul>	<p>Eine Liste der Einrichtungen mit unzureichender Personalsituation wird zurückgeliefert.</p> <table border="1"> <thead> <tr> <th colspan="2">Parameter Validation</th> </tr> </thead> <tbody> <tr> <td>traeger</td> <td><b>traegernr_type</b></td> </tr> </tbody> </table>	Parameter Validation		traeger	<b>traegernr_type</b>		
Parameter Validation									
traeger	<b>traegernr_type</b>								

*Tabelle 8 Anwendungsfall stellenangebot*

Anwendungsfall: <i>stellenangebot</i>									
Funktion	Methode	Parameter	Beschreibung						
lieferung	POST	–	Die Datenlieferung im Request-Body wird hochgeladen.						
abfragestatus	GET	<ul style="list-style-type: none"> <li>• traeger</li> <li>• einrichtung (optional)</li> </ul>	<p>Die Abfrage liefert den Status von Stellenangeboten für einen Träger bzw. eine Einrichtung zurück. Der Parametereinrichtung ist optional.</p> <table border="1"> <thead> <tr> <th colspan="2">Parameter Validation</th> </tr> </thead> <tbody> <tr> <td>traeger</td> <td><b>traegernr_type</b></td> </tr> <tr> <td>einrichtung</td> <td><b>einrichtungnr_type</b></td> </tr> </tbody> </table>	Parameter Validation		traeger	<b>traegernr_type</b>	einrichtung	<b>einrichtungnr_type</b>
Parameter Validation									
traeger	<b>traegernr_type</b>								
einrichtung	<b>einrichtungnr_type</b>								

*Fortsetzung auf der nächsten Seite*

*Tabelle 8: Anwendungsfall stellenangebot (Fortsetzung)*

Anwendungsfall: stellenangebot									
Funktion	Methode	Parameter	Beschreibung						
abfragedetail	GET	<ul style="list-style-type: none"> <li>• traeger</li> <li>• einrichtung</li> </ul>	liefert die Daten zu den Stellenangeboten einer Einrichtung. <table border="1" data-bbox="954 611 1310 734"> <thead> <tr> <th colspan="2">Parameter Validation</th> </tr> </thead> <tbody> <tr> <td>traeger</td> <td><b>traegernr_type</b></td> </tr> <tr> <td>einrichtung</td> <td><b>einrichtungnr_type</b></td> </tr> </tbody> </table>	Parameter Validation		traeger	<b>traegernr_type</b>	einrichtung	<b>einrichtungnr_type</b>
Parameter Validation									
traeger	<b>traegernr_type</b>								
einrichtung	<b>einrichtungnr_type</b>								

*Tabelle 9: Anwendungsfall vertrag*

Anwendungsfall: vertrag									
Funktion	Methode	Parameter	Beschreibung						
abfragedetail	GET	<ul style="list-style-type: none"> <li>• traeger</li> <li>• einrichtung</li> </ul>	liefert Vertragsinformationen zu einem Träger zurück. <table border="1" data-bbox="954 1081 1310 1205"> <thead> <tr> <th colspan="2">Parameter Validation</th> </tr> </thead> <tbody> <tr> <td>traeger</td> <td><b>traegernr_type</b></td> </tr> <tr> <td>einrichtung</td> <td><b>einrichtungnr_type</b></td> </tr> </tbody> </table>	Parameter Validation		traeger	<b>traegernr_type</b>	einrichtung	<b>einrichtungnr_type</b>
Parameter Validation									
traeger	<b>traegernr_type</b>								
einrichtung	<b>einrichtungnr_type</b>								

*Tabelle 10: Anwendungsfall vormerkung*

Anwendungsfall: vormerkung									
Funktion	Methode	Parameter	Beschreibung						
lieferung	POST	–	Die Datenlieferung im Request-Body wird hochgeladen.						
abfragestatus	GET	<ul style="list-style-type: none"> <li>• traeger</li> <li>• einrichtung (<i>optional</i>)</li> </ul>	Die Abfrage liefert den Status von Vormerkungen für einen Träger bzw. eine Einrichtung zurück. Der Parametereinrichtung ist optional. <table border="1" data-bbox="954 1731 1310 1854"> <thead> <tr> <th colspan="2">Parameter Validation</th> </tr> </thead> <tbody> <tr> <td>traeger</td> <td><b>traegernr_type</b></td> </tr> <tr> <td>einrichtung</td> <td><b>einrichtungnr_type</b></td> </tr> </tbody> </table>	Parameter Validation		traeger	<b>traegernr_type</b>	einrichtung	<b>einrichtungnr_type</b>
Parameter Validation									
traeger	<b>traegernr_type</b>								
einrichtung	<b>einrichtungnr_type</b>								

*Fortsetzung auf der nächsten Seite*

Tabelle 10: Anwendungsfall *vormerkung* (Fortsetzung)

Anwendungsfall: <i>vormerkung</i>									
Funktion	Methode	Parameter	Beschreibung						
abfragedetail	GET	<ul style="list-style-type: none"> <li>• traeger</li> <li>• einrichtung</li> </ul>	Eine Liste der Vormerkungen für die abgefragte Einrichtung wird zurückgeliefert. <table border="1" data-bbox="954 611 1310 741"> <thead> <tr> <th colspan="2">Parameter Validation</th> </tr> </thead> <tbody> <tr> <td>traeger</td> <td>traegernr_type</td> </tr> <tr> <td>einrichtung</td> <td>einrichtungnr_type</td> </tr> </tbody> </table>	Parameter Validation		traeger	traegernr_type	einrichtung	einrichtungnr_type
Parameter Validation									
traeger	traegernr_type								
einrichtung	einrichtungnr_type								

Grundsätzlich sind nur Kita-Träger und Kita-Einrichtungen als Aufruf-Parameter erlaubt (d. h. mit einer Träger-ID aus 4 Ziffern und einer Einrichtung-ID aus 8 Ziffern).

Bei der Funktion *lieferung* enthält jeder Datensatz eine *create*-, *update*- oder *delete*-Aktion. Ein Datensatz wird mit *create* neu angelegt, mit *update* bearbeitet und mit *delete* gelöscht. Im letzten Fall stehen keine Fachdaten im XML-Body.

Tabelle 11: Aktionen für die Funktion *lieferung*

Anwendungsfall	Aktion	Beschreibung
<ul style="list-style-type: none"> <li>• freiplatzmeldung</li> <li>• personalplanung</li> <li>• stellenangebot</li> <li>• vormerkung</li> </ul>	create	Datensatz wird neu erstellt.
	update	Datensatz wird bearbeitet.
	delete	Datensatz wird gelöscht.
<ul style="list-style-type: none"> <li>• kitaverzeichnis</li> </ul>	update	Datensatz wird bearbeitet.

#### 4.2.5.2 XSD-Schema technische Validierung bei Datenlieferung

Im Folgenden wird die technische Validierungen für die Datenlieferung der einzelnen Anwendungsfälle genauer betrachtet. Hierbei wird nicht auf Validierungen eingegangen, welche bereits aus dem XSD-Schema ersichtlich sind oder dem fachlichen Kontext unterliegen. Die maximale Länge (zb. "string255\_type" sind max 255 Zeichen) und ob ein Feld ein Pflichtfeld ist (zb. "minOccurs=" 1" ist ein Pflichtfeld) ist bereits Bestandteil des XSD-Schemas. Ist ein Freitextfeld nicht weiter erwähnt, ist keine zusätzliche Validierung des Datenwerts im Backend vorhanden.

Tabelle 12: Validierungen *personalplanung*

XSD-Type	Element	Validierung
personstamm_abstractType	<ul style="list-style-type: none"> <li>name</li> <li>vorname</li> </ul>	<p><b>Valide Zeichen:</b> Unicode Letter, Leerzeichen (SPACE), Punktionszeichen Punkt (FULL STOP) &amp; Minus (HYPHEN-MINUS)</p> <p><b>Regex:</b> /[\\p{L}\\u0020\\.-!{1,255}]/u <a href="http://12s://regexr.com/61267e">http://12s://regexr.com/61267e</a></p>
person_fachkraftType	<ul style="list-style-type: none"> <li>studienabschlussHochschule</li> <li>nameAbschluss</li> </ul>	<p><b>Invalide Zeichen:</b> ~#@*+%{}&lt;&gt; [ 11 " - " ! . ? =</p> <p><b>Regex:</b> [A~#@*+%{}&lt;&gt;\\[\\]   "\\_A!?=]{0,255} <a href="http://12s://regexr.com/6126cg">http://12s://regexr.com/6126cg</a></p>
	erlernterBeruf	<p><b>Valide Zeichen:</b> Unicode Letter &amp; Leerzeichen (SPACE)</p> <p><b>Regex:</b> /[\\p{L}\\u0020]{0,255}/u <a href="https://regexr.com/6p6kl">https://regexr.com/6p6kl</a></p>
person_fachkraftintegration_type	studienabschlussHochschule	<p><b>Invalide Zeichen:</b> ~#@*+%{}&lt;&gt; [ 11 " - " ! . ? =</p> <p><b>Regex:</b> [A~#@*+%{}&lt;&gt;\\[\\]   "\\_A!?=]{0,255} <a href="http://12s://regexr.com/6p6cg">http://12s://regexr.com/6p6cg</a></p>

#### 4.2.5.3 Identifikation von Person bei *personalme/dung*

Bei der Datenlieferung einer Person bei dem Anwendungsfall *personalplanung* ist es wichtig, dass eine Person eindeutig identifizierbar ist. Die Identifizierung von Personal findet auf Einrichtungsebene statt, d.h. eine Person wird im Backend mit der Zuordnung zu ihrer Einrichtung lokalisiert. Vielmehr ist ein Persondatensatz als ein Person-Einrichtung-Vertrag zu betrachten. Prinzipiell kann eine einzelne Person in mehreren Einrichtungen eingepflegt sein. Es liegt in der Verantwortung der Träger die Arbeitszeiten und Gültigkeiten bei Personalwechsel von Einrichtung A nach Einrichtung B entsprechend zu pflegen. Das bedeutet, dass der Träger hier zwei Datensätze für eine natürliche Person zu pflegen hat.

Bei der Implementierung von der Dienstschnittstelle *vorv1.18.0* wurde zur Identifizierung einer Person die */aufendeNr* aus dem XSD-Schema-Typ **personstamm\_abstractType** verwendet.

Seit *v1.18.0* wurde dies geändert hin zu einer Identifizierung mittels Kombination aus *name*, *vorname*, *geburtsdatum* aus dem XSD-Schema-Typ **personstamm\_abstract\_type**. Das Feld

laufendeNr ist nun kein Pflichtfeld mehr und wird vom Backend bei der Datenlieferung ignoriert.

#### 4.2.5.4 Abfrage von Statusinformationen einzelner Datensätze nach einer Lieferung

Für jede technisch fehlerfreie Lieferung wird durch die Dienstschnittstelle eine Vorgangsnummer (Trackingnummer) erzeugt. Über den Anwendungsfall *protoko/1* können die einzelnen Status der Lieferung abgefragt werden. In der Protokollantwort lässt sich über das Attribut lfdnummer im Tag <datensatz> der Datensatz aus der Lieferung identifizieren. Dabei kann ein Datensatz einen der folgenden Status haben:

*Tabelle 13:* Status einzelner Datensätze

Status	Erklärung
OK	Der Datensatz wurde von der Dienstschnittstelle angenommen und hat die fachliche Validierung im Trägerportal bestanden. Dieser Datensatz wird an die zuständige Fachanwendung in ISBJ weitergeleitet.
WARNING	Der Datensatz weist ein Problem auf, das eine Verarbeitung aber nicht verhindert (z. B. ein Update ohne eine Änderung). Dieser Datensatz wird ggf. an die zuständige Fachanwendung in ISBJ weitergeleitet. Details zu dem Problem werden in den Fehlermeldungen aufgeführt.
ERROR	Der Datensatz ist fehlerhaft und kann nicht verarbeitet werden. Details werden in den Fehlermeldungen aufgeführt.

Für die gesamte Lieferung steht ebenfalls eine Statusanzeige zur Verfügung. Diese dient der Entlastung externer Systeme, da die einzelnen Datensätze im OK-Fall nicht geprüft werden müssen.

*Tabelle 14:* Status einer Lieferung

Status	Erklärung
OK	Alle Datensätze in der Datenlieferung haben den Status OK.
WARNING	Nicht alle Datensätze in der Datenlieferung haben den Status OK. Hier muss im Einzelfall geprüft werden, welche Datensätze fehlerhaft sind.
ERROR	Die gesamte Datenlieferung ist fehlerhaft. Alle Datensätze in der Datenlieferung sind fehlerhaft.

## 4.2.6 Response-Codes

1 *1* **Tabelle 15:** Response-Codes

Response-Code	Nachricht	Bedeutung
200	OK	Die Daten wurden erfolgreich übermittelt bzw. die Request wurde verstanden und ohne Fehler durchgeführt. <i>Achtung: HTTP-200 wird auch für Ergebnisse verwendet, die fachliche Probleme haben (z. B. einzelne Datensätze werden abgelehnt, weil diese Duplikate sind).</i>
400	Bad Request	Die Clientanwendung hat ein Fehler gemacht, z. B. falsche HTTP-Methode, XML-Daten die nicht schemakonform sind usw.
407	Unauthorized	Das Zertifikat ist nicht für diesen Anwender. Benutzername oder Kennwort sind falsch oder der Basic-Authentication Header ist nicht gesetzt bzw. nicht lesbar.
403	Forbidden	Keine Berechtigung für diese Aktion, obwohl die Anmeldung erfolgreich war. Z. B. Datenabfrage für einen Träger, für den der Benutzer keine Rechte hat.
404	Not Found	1 Falsche URL
500	Internal Server Error	In der Dienstschnittstelle ist ein unbehandelter Fehler aufgetreten. Dieser Fehler muss zusammen mit der Uhrzeit, dem Benutzernamen und weiteren sinnvollen Details an den entsprechenden Helpdesk oder dem Träger-Service gemeldet werden.
503	Service Unavailable	In der Dienstschnittstelle bzw. dem Trägerportal stehen Wartungsarbeiten an und die Anwendung ist nicht verfügbar.

## 4.2.7 Eigenschaften der Dienstschnittstelle

### 4.2.7.1 Datenlieferung

Ab der Version 1.3.0 der Dienstschnittstelle ist es nicht mehr vorgesehen, dass vollständige Lieferungen des extern gehaltenen Datenbestandes an die ISBJ-Dienstschnittstelle vorgenommen werden. Es ist zwar weiterhin möglich, vollständige Lieferungen zu tätigen, um alle beteiligten Systeme zu entlasten, wird jedoch davon abgeraten. Stattdessen wird empfohlen, nur die geänderten Datensätze zu liefern.

#### 4.2.7.2 Reaktionszeiten auf verschiedene Aktionen bei Lieferungen

Grundsätzlich können Datensätze, die über die Dienstschnittstelle angelegt oder bearbeitet werden, erst dann erneut geändert werden, nachdem sie von der dem Anwendungsfall zugehörigen Backendkomponente freigegeben wurden. Dies kann je nach angegebener Aktion und Größe der gelieferten Datei sowie der allgemeinen Last auf den ISBJ-Systemen mehrere Minuten dauern. Die Aktion *delete* stellt hierbei einen Sonderfall dar.

Die folgende Tabelle gibt eine Übersicht über das erwartete Verhalten der Aktionen:

*Tabelle 16:* Reaktionsverhalten auf Aktionen bei Lieferungen

Aktion	Verhalten
<i>create &amp; update</i>	Die Datensätze sollten innerhalb weniger Minuten über die Dienstschnittstelle auslesbar und erneut bearbeitbar sein.
<i>delete</i>	Gelöschte Datensätze werden im Trägerportal als <i>in Bearbeitung</i> markiert und einmal in 24 Stunden im Rahmen des Datenabgleichs zwischen Trägerportal und Backendkomponente aus dem Datenbestand des Trägerportals entfernt.

#### 4.2.7.3 Zusätzliche Informationen zum XSD-Schema

Mit dem Schema werden aus folgenden Gründen Datentypen definiert und Einschränkungen vorgenommen:

##### Datumstypen

Es existieren zwei Standarddatumstypen:

- *xs:date* wird in der XML-Datei repräsentiert durch `<datum>yyyy-MM-dd</datum>`
- *xs:dateTime* wird in der XML-Datei repräsentiert durch `<datum>yyyy-MM-ddThh mm ss</datum>`

Diese Typen wurden gewählt, um die Kommunikation aller am Fachverfahren beteiligten Komponenten sicherzustellen.

##### Definition des Geburtsdatums bei Vormerkung

Für die erfolgreiche Verarbeitung der Datenlieferung ist es zwingend erforderlich für das Geburtsdatum neben dem Geburtsjahr auch Tag und Monat anzugeben, da alle ISBJ-Anwendungen bisher nur mit vollständigem Geburtsdatum arbeiten können.

##### Einschränkungen der Datensätze innerhalb einer Lieferung

Derzeit ist eine Datenlieferung durch folgende Kardinalitäten begrenzt:

- 1 Träger pro Lieferung
- 200 Einrichtungen pro Träger
- 1000 Datensätze pro Einrichtung

Diese Einschränkungen wurden unter Berücksichtigung der derzeit in der Praxis vorkommenden Anwendungsfälle festgelegt. Sollten diese Begrenzungen überschritten werden, muss der Träger-Service über diesen Anwendungsfall informiert werden. Hierbei sei auf Kapitel 4.2.7.1 verwiesen. Vollständige Updates sind nicht mehr vorgesehen und können ggf. in mehrere kleinere Lieferungen unterteilt werden.

#### **Einschränkung der Wertebereiche**

Die Dienstschnittstelle akzeptiert nur eingeschränkte Wertebereiche, die durch die Längendefinitionen der Datenbankschemata in anderen Komponenten vorgegeben sind. Anwendungsfälle, die diese Einschränkungen überschreiten, sind dem Träger-Service mitzuteilen.

#### **4.2.7.4 Fehlermeldungen und deren Weitergabe**

Grundsätzlich muss entschieden werden, ob der Fehler technische oder fachliche Ursachen hat. Fehlermeldungen, die darauf hindeuten, dass die Daten nicht schemakonform sind, kann kein Anwender beheben. Diese und weitere technischen Fehler müssen an die Entwickler der externen Anwendung weitergeleitet werden. Fachliche Fehlermeldungen sollten deshalb dem Anwender im externen System dargestellt werden, sodass die Daten korrigiert werden können.

Es wird dringend empfohlen, Fehlermeldungstexte aus der Dienstschnittstelle nicht vor der Darstellung zu verändern, anderenfalls wird eine Überprüfung seitens des Träger-Services oder der Nortal AG erschwert.

Sollte eine Anfrage an die Dienstschnittstelle gerichtet werden, bei der im Header ein falscher Nutzername oder ein falsches Passwort übergeben wird, erstellt die Dienstschnittstelle zwei unterschiedliche Fehlermeldungen. Dies ist ein gewünschtes Verhalten. Sofern die Fehlermeldungen im externen System gleichgestellt werden sollen, können über den folgenden regulären Ausdruck beide Fehlermeldungen ausgewertet werden:

```
A(Der Benutzer).( oder das Passwort).( falsch\.)$
```