

Schnittstellenspezifikation

# SchulConneX

Beschreibung der Schnittstellen von SchulConneX für Dienste und Quellsysteme

---

**Version 1.004.042.000** (Veröffentlichungskandidat )

*[Interner Arbeitsentwurf | Kommentierbarer Arbeitsentwurf | Veröffentlichungskandidat | Veröffentlichte Spezifikation]*

Stand vom 11.01.2024

Aufgabensteller:

Landesinitiative n-21: Schulen in Niedersachsen online e. V.

Dr. Boris Heithecker (Ansprechpartner)

Schiffgraben 27

30159 Hannover

Fachliche Ausarbeitung der Schnittstellenspezifikation:

Fraunhofer-Institut für Offene Kommunikationssysteme FOKUS

Dr. Christopher Krauß (Ansprechpartner)

Kaiserin-Augusta-Allee 31

10589 Berlin | Deutschland

Landesinitiative n-21:  
Schulen in Niedersachsen online



im Auftrag des Niedersächsischen  
Kultusministeriums

## **Rechtliche Hinweise**

Dieses Werk ist lizenziert unter [Creative Commons Namensnennung-Keine Bearbeitung 2.0 Deutschland \(CC BY-ND 2.0 DE\)](https://creativecommons.org/licenses/by-nd/2.0/de/).

Das Dokument wurde nach bestem Wissen und mit größter Sorgfalt erstellt, dennoch können Fehler und Ungenauigkeiten nicht vollständig ausgeschlossen werden. Eine juristische Verantwortung oder Haftung für eventuell verbliebene fehlerhafte Angaben und deren Folgen wird nicht übernommen.

# Inhaltsverzeichnis

1	STAND DER BEARBEITUNG	18
2	VORBEMERKUNG	18
3	EINLEITUNG	18
4	ALLGEMEINES	19
4.1	Kernanwendungsfälle	19
4.2	Nutzung der Schnittstelle durch Dienste	19
4.3	Ablauf Löschen von Personenkontexten und Personen	19
4.3.1	Löschen von Personenkontexten und Personen aus Sicht des primären Quellsystems	20
4.3.2	Information von Diensten über die Löschung von Personenkontexten	20
5	DATENMODELL DER SCHNITTSTELLE	21
5.1	Datumsformat	22
5.2	Organisation	22
5.3	Person	24
5.4	Personenkontext	31
5.5	Personendatensatz	34
5.6	Gruppe	37
5.7	Gruppenzugehörigkeit	41
5.8	Gruppendatensatz	43
5.9	Lernperiode	45
5.10	Kommunikation	47
5.11	Beziehung	48
5.12	Anschrift	49
5.13	Organisationsbeziehung	49
6	ANGEPASSTES DATENMODELL FÜR DIENSTANBIETER-SCHNITTSTELLE	51

6.1	Angepasstes Datenmodell Person für Dienstanbieter	53
6.2	Angepasstes Datenmodell Personenkontext für Dienstanbieter	55
6.3	Angepasstes Datenmodell Organisation für Dienstanbieter	58
6.4	Angepasstes Datenmodell Gruppe für Dienstanbieter	59
7	SCHNITTSTELLEDEFINITION	62
7.1	Allgemeine Festlegungen	62
7.1.1	Standard-Anfragen (Default Requests)	62
7.1.2	Standard-Rückgaben (Default Responses)	62
7.2	Erfolgreiche Ausführung	63
7.3	Daten unverändert	63
7.4	Fehlerbehandlung	63
7.5	Validierung der Anfrage	69
8	SCHNITTSTELLEN FÜR QUELLSYSTEME UND DIENSTE	70
8.1	API-Versionierung	70
8.2	Registrierung	70
8.3	API-Autorisierung für Quellsysteme	71
8.4	Autorisierung und Authentifikation bei Dienstanbietern	71
8.4.1	Access Token	73
8.4.2	ID Token	75
8.5	Vorbedingungen der Schnittstelle	76
9	API-ENDPUNKTE FÜR DIENSTANBIETER	78
9.1	Schnittstellenendpunkte für Personen /person-info	78
9.1.1	Schnittstelle „person-info“	78
9.2	Person-Info über OIDC Claims	85
9.3	Schnittstellenendpunkte für Personen /personen-info	86
9.3.1	Schnittstelle „personen-info“	86
10	API-ENDPUNKTE FÜR QUELLSYSTEME	98
10.1	Schnittstellen für Personen	98
10.1.1	[/personen] Personendatensätze erstellen und auflisten	98
10.1.2	[/personen/{id}] Individuelle Personendatensätze lesen, ändern und löschen	108

10.1.3	<b>[/personen/{id}/personenkontexte]</b>	Personenkontexte lesen und erstellen	118
10.1.4	<b>[/personen/{id}/erreichbarkeiten]</b>	Erstellen und Lesen von zu einer Person gehörenden Erreichbarkeiten	125
10.1.5	<b>[/personenkontexte]</b>	Personenkontexte der eigenen Organisation lesen	131
10.1.6	<b>[/personenkontexte/{id}]</b>	Individuelle Personenkontexte lesen, ändern und löschen	137
10.1.7	<b>[/personenkontexte/{personenkontext.id}/beziehungen]</b>	Erstellen und Lesen von zu einem Personenkontext gehörenden Beziehungen	146
10.1.8	<b>[/beziehungen/{beziehung.id}]</b>	Lesen und löschen einer einzelnen Beziehung	154
10.1.9	<b>[/personenkontexte/{personenkontext.id}/erreichbarkeiten]</b>	Erstellen und Lesen von zu einem Personenkontext gehörenden Erreichbarkeiten	156
10.1.10	<b>[/erreichbarkeiten/{erreichbarkeit.id}]</b>	Lesen und löschen einer einzelnen Erreichbarkeit	161
<b>10.2</b>	<b>Schnittstellen für Organisationen</b>		<b>164</b>
10.2.1	<b>[/organisationen]</b>	Organisationen auflisten	164
10.2.2	<b>[/organisationen/{id}]</b>	Lesen einer individuellen Organisation	168
10.2.3	<b>[/organisationen/{id}/organisationsbeziehungen]</b>	Lesen von zu einer Organisation gehörenden Organisationsbeziehungen	170
10.2.4	<b>[/organisation-info]</b>	Lesen der eigenen Organisation	175
<b>10.3</b>	<b>Schnittstellen für Gruppen</b>		<b>178</b>
10.3.1	<b>[/gruppen]</b>	Gruppen erstellen und lesen	178
10.3.2	<b>[/gruppen/{id}]</b>	Lesen, ändern und löschen einer individuellen Gruppe	189
10.3.3	<b>[/gruppen/{gruppe.id}/gruppenzugehoerigkeiten]</b>	Erstellen und Lesen von zu einer Gruppe gehörenden Gruppenzugehörigkeiten	199
10.3.4	<b>[/gruppenzugehoerigkeiten]</b>	Gruppenzugehörigkeiten der eigenen Organisation lesen	205
10.3.5	<b>[/gruppenzugehoerigkeiten/{gruppenzugehoerigkeit.id}]</b>	Lesen, ändern und löschen einer individuellen Gruppenzugehörigkeit	209
<b>11</b>	<b>ALLGEMEINE API-ENDPUNKTE</b>		<b>217</b>
<b>11.1</b>	<b>Schnittstellen für Codelisten</b>		<b>217</b>
11.1.1	<b>[/codelisten]</b>	Auflistung verfügbarer Codelisten	217
11.1.2	<b>[/codelisten/{codeliste}]</b>	Auflistung des Inhalts einer Codeliste	218
<b>11.2</b>	<b>Schnittstellen für die Versionsverwaltung</b>		<b>218</b>
11.2.1	<b>[/versionen]</b>	Auflistung verfügbarer Codelisten	218
<b>12</b>	<b>CODELISTEN</b>		<b>221</b>
<b>12.1</b>	<b>Personenstatus</b>		<b>221</b>
<b>12.2</b>	<b>Geschlecht</b>		<b>221</b>
<b>12.3</b>	<b>Rolle</b>		<b>222</b>
<b>12.4</b>	<b>Vertrauensstufe</b>		<b>222</b>
<b>12.5</b>	<b>Organisationstyp</b>		<b>223</b>
<b>12.6</b>	<b>Jahrgangsstufe</b>		<b>223</b>

<b>12.7</b>	<b>Boolean</b>	<b>224</b>
<b>12.8</b>	<b>Trägerschaft</b>	<b>224</b>
<b>12.9</b>	<b>Lokalisierung</b>	<b>224</b>
<b>12.10</b>	<b>Gruppenbereich</b>	<b>225</b>
<b>12.11</b>	<b>Gruppendifferenzierung</b>	<b>225</b>
<b>12.12</b>	<b>Gruppenoption</b>	<b>225</b>
<b>12.13</b>	<b>Gruppentyp</b>	<b>226</b>
<b>12.14</b>	<b>Gruppenrolle</b>	<b>226</b>
<b>12.15</b>	<b>Lernperiode</b>	<b>226</b>
<b>12.16</b>	<b>Lernperiodentyp</b>	<b>227</b>
<b>12.17</b>	<b>Fächerkanon</b>	<b>227</b>
<b>12.18</b>	<b>Bildungsziel</b>	<b>229</b>
<b>12.19</b>	<b>Erreichbarkeitstyp</b>	<b>229</b>
<b>12.20</b>	<b>Beziehungen</b>	<b>230</b>
<b>12.21</b>	<b>Organisationsbeziehungen</b>	<b>230</b>
<b>13</b>	<b>GLOSSAR</b>	<b>232</b>
<b>14</b>	<b>ENGLISH API NOTES (ENG.)</b>	<b>234</b>
<b>14.1</b>	<b>Data Models (eng.)</b>	<b>234</b>
14.1.1	Format of dates (eng.)	234
14.1.2	Organisation (eng.)	234
14.1.3	Person (eng.)	235
14.1.4	Personenkontext (eng.)	238
14.1.5	Personendatensatz (eng.)	239
14.1.6	Gruppe (eng.)	239
14.1.7	Gruppenzugehörigkeit (eng.)	241
14.1.8	Gruppendatensatz (eng.)	242
14.1.9	Lernperiode (eng.)	243
14.1.10	Kommunikation (eng.)	243
14.1.11	Beziehung (eng.)	244
14.1.12	Anschrift (eng.)	244
14.1.13	Organisationsbeziehung (eng.)	244
<b>14.2</b>	<b>Code lists (eng.)</b>	<b>246</b>
<b>15</b>	<b>PRAXISLEITFADEN FÜR DIE IMPLEMENTIERUNG</b>	<b>248</b>

<b>15.1</b>	<b>Ablaufbeschreibung: Synchronisation mit dem primären Quellsystem</b>	<b>248</b>
15.1.1	Konfiguration	248
15.1.2	Synchronisation	248
<b>15.2</b>	<b>Ablauf des Schuljahreswechsels für Quellsysteme</b>	<b>255</b>
15.2.1	Besonderheiten bei der Aktualisierung der Personen und Personenkontexte	255
15.2.2	Besonderheiten beim Aktualisieren von Gruppen	256
<b>15.3</b>	<b>Änderung der Rolle in einer Organisation</b>	<b>258</b>
<b>16</b>	<b>ANLAGE: DOKUMENTHISTORIE</b>	<b>260</b>

## Abbildungsverzeichnis

Abbildung 1: Graphische Darstellung des Datenmodells	22
Abbildung 2: Graphische Darstellung des vereinfachten Datenmodells für Dienste	52
Abbildung 3: Sequenzdiagramm für die Autorisierung und anschließende Beispielabfrage von Personendaten im Sicherheitskontext einer angemeldeten nutzenden Person	72

## Datenmodelle

Datenmodell 1: Datenmodell einer Organisation	24
Datenmodell 2: Datenmodell einer Person	29
Datenmodell 3: Datenmodell eines Personenkontexts	33
Datenmodell 4: Datenmodell des Personendatensatzes	35
Datenmodell 5: Datenmodell der Gruppe	41
Datenmodell 6: Datenmodell der Gruppenzugehörigkeit	43
Datenmodell 7: Datenmodell des Gruppendatensatzes	43
Datenmodell 8: Datenmodell der Lernperiode	46
Datenmodell 9: Datenmodell Kommunikation	47
Datenmodell 10: Datenmodell Beziehung	48

Datenmodell 11: Datenmodell Anschrift	49
Datenmodell 12: Datenmodell Organisationsbeziehung	50
Datenmodell 13: Datenmodell Person für Dienstleister	55
Datenmodell 14: Datenmodell Personenkontext für Dienstleister	57
Datenmodell 15: Datenmodell Organisation für Dienstleister	58
Datenmodell 16: Angepasstes Datenmodell Gruppe für Dienstleister	61
Datenmodell 17: Access Token	75
Datenmodell 18: ID-Token	76

## Fehlercodelisten

Fehlerliste 1: Fehler 400 Bad Request	66
Fehlerliste 2: Fehler 401 Unauthorized	67
Fehlerliste 3: Fehler 403 Forbidden	67
Fehlerliste 4: Fehler 404 Not found	68
Fehlerliste 5: Fehler 405 Method not allowed	68
Fehlerliste 6: Fehler 409 Conflict	68
Fehlerliste 7: Fehler 500 Interner Serverfehler	69



# Tabellenverzeichnis

Tabelle 1: Zugelassene Operation auf dem Endpunkt „person-info“	78
Tabelle 2: Mapping von Attributen auf OIDC-Claim Namen	85
Tabelle 3: Zugelassene Operation auf dem Endpunkt „personen-info“	87
Tabelle 4: Zugelassene Filter auf dem Endpunkt /v1/personen-info	89
Tabelle 5: Zugelassene Operationen auf dem Endpunkt /v1/personen	98
Tabelle 6: Zugelassene Filter auf dem Endpunkt /v1/personen	99
Tabelle 7: Liste von Attributen, welche durch ein Quellsystem oder Dienst über den Endpunkt /v1/personen auf HTTP-POST nicht gesetzt werden können.	100
Tabelle 8: Zugelassene Operationen auf dem Endpunkt /v1/personen/{personendatensatz.person.id}	108
Tabelle 9: Liste von Attributen, welche durch ein Quellsystem oder Dienst über den Endpunkt /v1/personen/{personendatensatz.person.id} auf HTTP-PUT nicht verändert werden können.	112
Tabelle 10: Zugelassene Operationen auf dem Endpunkt /v1/personen/{personendatensatz.person.id}/personenkontexte	118
Tabelle 11: Zugelassene Filter auf dem Endpunkt /v1/personen/{personendatensatz.person.id}/personenkontexte	119
Tabelle 12: Liste von Attributen, welche durch ein Quellsystem oder Dienst über den Endpunkt /v1/personen/{personendatensatz.person.id}/personenkontexte auf HTTP-POST nicht gesetzt werden können.	121
Tabelle 13: Zugelassene Operationen auf dem Endpunkt /v1/personen/{id}/erreichbarkeiten	126
Tabelle 14: Liste von Attributen, welche durch ein Quellsystem oder Dienst über den Endpunkt /v1/personen/{id} /erreichbarkeiten auf HTTP-POST nicht gesetzt werden können.	127
Tabelle 15: Zugelassene Operation auf dem Endpunkt /v1/personenkontexte	131
Tabelle 16: Zugelassene Filter auf dem Endpunkt /v1/personenkontexte	132
Tabelle 17: Zugelassene Operationen auf dem Endpunkt /v1/personenkontexte/{personendatensatz.personenkontext.id}	137
Tabelle 18: Liste von Attributen, welche durch ein Quellsystem oder Dienst über den Endpunkt /v1/personenkontexte/{personendatensatz.personenkontext.id} auf HTTP-PUT nicht verändert werden können.	142
Tabelle 19: Zugelassene Operationen auf dem Endpunkt /v1/personenkontext/{personenkontext.id}/beziehungen	147

Tabelle 20: Liste von Attributen, welche durch ein Quellsystem oder Dienst über den Endpunkt /v1/personenkontexte/{personenkontext.id} /bezugskontext auf HTTP-POST nicht gesetzt werden können.	148
Tabelle 21: Zugelassene Filter auf dem Endpunkt /v1/personenkontexte/{personenkontext.id}/beziehungen	151
Tabelle 22: Zugelassene Operationen auf dem Endpunkt /v1/beziehungen/{beziehung.id}	154
Tabelle 23: Zugelassene Operationen auf dem Endpunkt /v1/personenkontext/{personenkontext.id}/erreichbarkeiten	157
Tabelle 24: Liste von Attributen, welche durch ein Quellsystem oder Dienst über den Endpunkt /v1/personenkontexte/{personenkontext.id} /erreichbarkeiten auf HTTP-POST nicht gesetzt werden können.	158
Tabelle 25: Zugelassene Operationen auf dem Endpunkt /v1/erreichbarkeiten/{erreichbarkeit.id}	162
Tabelle 26: Zugelassene Operation auf dem Endpunkt /v1/organisationen	164
Tabelle 27: Zugelassene Filter auf dem Endpunkt /v1/organisationen	165
Tabelle 28: Zugelassene Operation auf dem Endpunkt /v1/organisationen/ {organisation.id}	168
Tabelle 29: Zugelassene Operation auf dem Endpunkt /v1/organisationen/{id}/organisationsbeziehungen	170
Tabelle 30: Zugelassene Filter auf dem Endpunkt /v1/organisationen/{id}/organisationsbeziehungen	171
Tabelle 31: Zugelassene Operation auf dem Endpunkt /v1/organisation-info	175
Tabelle 32: Zugelassene Operationen auf dem Endpunkt /v1/gruppen	178
Tabelle 33: Zugelassene Filter auf dem Endpunkt /v1/gruppen	183
Tabelle 34: Zugelassene Operationen auf dem Endpunkt /v1/gruppen/ {gruppe.id}	189
Tabelle 35: Liste von Attributen, welche durch ein Quellsystem oder Dienst über den Endpunkt /v1/gruppen/{gruppe.id} auf HTTP-PUT nicht verändert werden können.	194
Tabelle 36: Zugelassene Operationen auf dem Endpunkt /v1/gruppen/{gruppe.id}/gruppenzugehoerigkeiten	199
Tabelle 37: Liste von Attributen, welche durch ein Quellsystem oder Dienst über den Endpunkt /v1/gruppen/{gruppe.id} /gruppenzugehoerigkeiten auf HTTP-POST nicht gesetzt werden können.	200
Tabelle 38: Zugelassene Filter auf dem Endpunkt /v1/gruppen/{gruppe.id}/gruppenzugehoerigkeiten /v1/gruppen	203
Tabelle 39: Zugelassene Operation auf dem Endpunkt /v1/gruppenzugehoerigkeiten	205
Tabelle 40: Zugelassene Filter auf dem Endpunkt /v1/gruppenzugehoerigkeiten	206

Tabelle 41: Zugelassene Operationen auf dem Endpunkt /v1/gruppenzugehoerigkeiten/{gruppenzugehoerigkeit.id}	209
Tabelle 42: Liste von Attributen, welche durch ein Quellsystem oder Dienst über den Endpunkt /v1/gruppenzugehoerigkeit/{gruppenzugehoerigkeit.id} auf HTTP-PUT nicht verändert werden können.	213
Tabelle 43: Zugelassene Operation auf dem Endpunkt /v1/codelisten	217
Tabelle 44: Zugelassene Operation auf dem Endpunkt /v1/codelisten/{codeliste}	218
Tabelle 45: Zugelassene Operation auf dem Endpunkt /v1/versionen	219
Tabelle 46: Ablaufbeschreibung für den Abgleich der Datensätze	251
Tabelle 47: Ablaufbeschreibung für das Aktualisieren und Anlegen der Datensätze	251
Tabelle 48: Ablaufbeschreibung für den Abgleich der Datensätze (Gruppen und Gruppenzugehörigkeiten)	253
Tabelle 49: Ablaufbeschreibung für das Aktualisieren und Anlegen der Datensätze (Gruppen und Gruppenzugehörigkeiten)	254
Tabelle 50: Besonderheiten beim Aktualisieren fortgeführter Gruppen	257
Tabelle 51: Besonderheiten beim Aktualisieren beendeter Gruppen	257

## Quellcodes

Quellcode 1: Beispiel JSON für eine Organisation	24
Quellcode 2: Beispiel JSON einer Person	30
Quellcode 3: Beispiel JSON eines Personenkontexts	34
Quellcode 4: Beispiel JSON eines Personendatensatzes	36
Quellcode 5: Beispiel JSON eines Gruppendatensatzes	45
Quellcode 6: Beispiel JSON des Datentyps Kommunikation	47
Quellcode 7: Beispiel JSON für eine Beziehung	48
Quellcode 8: Beispiel JSON für eine Anschrift	49
Quellcode 9: Beispiel JSON für eine Organisationsbeziehung	50
Quellcode 10: Beispiel des Error Payloads	63
Quellcode 11: URL-Struktur der Schnittstelle	70
Quellcode 12: Beispiel der URL eines Schnittstellenendpunkts	70
Quellcode 13: Beispiel eines Authorization Headers	77
Quellcode 14: Beispiel eines Accept Headers	77
Quellcode 15: Beispiel des Schnittstellen-Endpunkts für person-info	78

Quellcode 16: JSON-Struktur der Rückgabe zu person-info	80
Quellcode 17: Beispiel einer JSON-Rückgabe zu person-info	84
Quellcode 18: Beispiel einer JSON-Rückgabe mit OIDC Claims ohne Scope Request	86
Quellcode 19: Beispiel einer JSON-Rückgabe mit Scope Requests für <code>person-info</code>	86
Quellcode 20: Beispiel des Schnittstellen-Endpunkts für personen-info	86
Quellcode 21: JSON-Struktur der Rückgabe zu personen-info	90
Quellcode 22: Beispiel einer JSON-Rückgabe zu personen-info	90
Quellcode 23: Beispiel einer JSON-Rückgabe zu personen-info mit Filterparametern 'vollstaendig'	95
Quellcode 24: Beispiel einer JSON-Rückgabe zu personen-info mit Filterparametern 'vollstaendig' und Auswahl nach Gruppe	97
Quellcode 25: URL-Struktur des Endpunkts personen	98
Quellcode 26: JSON-Struktur eines CREATE-Requests an <code>/v1/personen</code>	101
Quellcode 27: Beispiel Request zu <code>/personen</code> auf HTTP-POST	102
Quellcode 28: JSON-Struktur einer CREATE-Response an <code>/v1/personen</code>	102
Quellcode 29: Beispiel Response zu <code>/personen</code> auf HTTP-POST	103
Quellcode 30: JSON-Struktur eines READ-Requests an <code>/v1/personen</code>	104
Quellcode 31: Beispiel Response zu <code>/personen</code> auf HTTP-GET	106
Quellcode 32: Beispiel Response zu <code>/personen</code> auf HTTP-GET mit den Filter-parametern (query-parameter) „?referrer=123“	107
Quellcode 33: URL-Struktur des Endpunkts <code>personen/{personendatensatz.person.id}</code>	108
Quellcode 34: JSON-Struktur der Antwort bei READ auf <code>/v1/personen/{personendatensatz.person.id}</code>	109
Quellcode 35: Beispiel Response zu <code>/personen/{personendatensatz.person.id}</code> auf HTTP-GET	110
Quellcode 36: JSON-Struktur eines READ-Requests an <code>/v1/personen/{personendatensatz.person.id}</code>	112
Quellcode 37: Spezielle JSON-Struktur eines READ-Requests an <code>/v1/personen/{personendatensatz.person.id}</code>	113
Quellcode 38: Beispiel Request zu <code>/personen/{personendatensatz.person.id}</code> auf HTTP-PUT	114
Quellcode 39: Beispiel Request zu <code>/personen/{personendatensatz.person.id}</code> auf HTTP-PUT	114
Quellcode 40: Beispiel Response zu <code>/personen/{personendatensatz.person.id}</code> auf HTTP-PUT	115

Quellcode 41: JSON-Struktur eines DELETE-Requests an /v1/personen/{personendatensatz.person.id}	116
Quellcode 42: Beispiel Request zu /personen/{personendatensatz.person.id} auf HTTP-DELETE	116
Quellcode 43: URL-Struktur des Endpunkts personen/{personendatensatz.person.id}/personenkontexte	118
Quellcode 44: JSON-Struktur eines CREATE-Requests an /v1/personen/{personendatensatz.person.id}/personenkontexte	121
Quellcode 45: Beispiel Request zu /personen/{personendatensatz.person.id}/personenkontexte auf HTTP-POST	121
Quellcode 46: JSON-Struktur einer CREATE-Response an /v1/personen/{personendatensatz.person.id}/personenkontexte	122
Quellcode 47: Beispiel Response zu /personen/{personendatensatz.person.id}/personenkontexte auf HTTP-POST	123
Quellcode 48: JSON-Struktur einer READ-Response an /v1/personen/{personendatensatz.person.id}/personenkontexte	123
Quellcode 49: Beispiel Response zu /personen/{personendatensatz.person.id}/personenkontexte auf HTTP-GET	124
Quellcode 50: Beispiel Response zu /personen/{personendatensatz.person.id}/personenkontexte auf HTTP-GET mit den Filter- parametern (query-parameter) „?referrer=NI_56-78-90_123“	125
Quellcode 51: URL-Struktur des Endpunkts /v1/personen/{id}/erreichbarkeiten	125
Quellcode 52: JSON-Struktur eines CREATE-Requests an /v1/personen/{id}/erreichbarkeiten	127
Quellcode 53: Beispiel Request zu /v1/personen/{id}/erreichbarkeiten auf HTTP-POST	128
Quellcode 54: JSON-Struktur einer CREATE-Response an /v1/personen/{id}/erreichbarkeiten	128
Quellcode 55: Beispiel Response zu /v1/personen/{id}/erreichbarkeiten auf HTTP-POST	128
Quellcode 56: JSON-Struktur einer READ-Response an /v1/ personen/{id}/erreichbarkeiten	129
Quellcode 57: Beispiel Response zu /personen/{id}/erreichbarkeiten	130
Quellcode 58: URL-Struktur des Endpunkts personenkontexte	131
Quellcode 59: JSON-Struktur einer READ-Response an /v1/personenkontexte	134
Quellcode 60: Beispiel Response zu /personenkontexte auf HTTP-GET	135
Quellcode 61: Beispiel Response zu personenkontexte auf HTTP-GET mit den Filter- parametern (query-parameter) „?referrer=NI_56789_123“	135

<i>Quellcode 62: URL-Struktur des Endpunkts</i>	
<i>personenkontexte/{personendatensatz.personenkontext.id}</i>	137
Quellcode 63: JSON-Struktur der Antwort bei READ auf <i>/ {personendatensatz.person.id}</i>	138
Quellcode 64: JSON-Struktur einer READ-Response an <i>/v1/personenkontexte/{id}</i>	138
Quellcode 65: Beispiel Response zu <i>/personenkontexte/{personendatensatz.personenkontext.id}</i> auf HTTP-GET	140
Quellcode 66: JSON-Struktur eines UPDATE-Requests an <i>/v1/personenkontexte/{id}</i>	143
Quellcode 67: Beispiel Request zu <i>/personenkontexte/{personendatensatz.personenkontext.id}</i> auf HTTP-PUT	143
Quellcode 68: JSON-Struktur einer UPDATE-Response an <i>/v1/personenkontexte/{id}</i>	144
Quellcode 69: Beispiel Response zu <i>/personenkontexte/{personendatensatz.personenkontext.id}</i> auf HTTP-PUT	144
<i>Quellcode 70: URL-Struktur des Endpunkts</i>	
<i>personenkontext/{personenkontext.id}/beziehungen</i>	146
Quellcode 71: JSON-Struktur eines CREATE-Requests an <i>/v1/personenkontexte/{personenkontext.id}/beziehungen</i>	149
Quellcode 72: Beispiel Request zu <i>/v1/personenkontexte/{personenkontext.id}/beziehungen</i> auf HTTP-POST	149
Quellcode 73: JSON-Struktur einer CREATE-Response an <i>/v1/personenkontexte/{personenkontext.id}/beziehungen</i>	149
Quellcode 74: Beispiel Response zu <i>/v1/personenkontexte/{personenkontext.id}/beziehungen</i> auf HTTP-POST	150
Quellcode 75: JSON-Struktur einer READ-Response an <i>/v1/ personenkontexte/{personenkontext.id}/beziehungen</i>	152
Quellcode 76: Beispiel Response zu <i>/personenkontext/{personenkontext.id}/beziehungen</i> auf HTTP-GET ohne Filter	153
Quellcode 77: Beispiel Response zu <i>/personenkontext/{personenkontext.id}/beziehungen</i> auf HTTP-GET mit Filter <i>"?hat_als_beziehungen=ja&amp;ist_von_beziehungen=ja"</i>	153
<i>Quellcode 78: URL-Struktur des Endpunkts</i> <i>/v1/beziehungen/{beziehung.id}</i>	154
Quellcode 79: JSON-Struktur einer READ-Response an <i>/v1/beziehungen/{beziehung.id}</i>	155
Quellcode 80: Beispiel Response zu <i>/v1/beziehungen/{beziehung.id}</i> auf HTTP-GET	155
Quellcode 81: JSON-Struktur eines DELETE-Requests an <i>/v1/beziehungen/{beziehung.id}</i>	156
Quellcode 82: Beispiel Request zu <i>/v1/beziehungen/{beziehung.id}</i> auf HTTP-DELETE	156
<i>Quellcode 83: URL-Struktur des Endpunkts</i>	
<i>personenkontext/{personenkontext.id}/erreichbarkeiten</i>	157

Quellcode 84: JSON-Struktur eines CREATE-Requests an /v1/personenkontexte/{personenkontext.id}/erreichbarkeiten	159
Quellcode 85: Beispiel Request zu /v1/personenkontexte/{personenkontext.id}/erreichbarkeiten auf HTTP-POST	159
Quellcode 86: JSON-Struktur einer CREATE-Response an /v1/personenkontexte/{personenkontext.id}/erreichbarkeiten	159
Quellcode 87: Beispiel Response zu /v1/personenkontexte/{personenkontext.id}/erreichbarkeiten auf HTTP-POST	159
Quellcode 88: JSON-Struktur einer READ-Response an /v1/ personenkontexte/ {personenkontext.id}/erreichbarkeiten	160
Quellcode 89: Beispiel Response zu /personenkontext/{personenkontext.id}/erreichbarkeiten	161
<i>Quellcode 90: URL-Struktur des Endpunkts /v1/erreichbarkeiten/{erreichbarkeit.id}</i>	161
Quellcode 91: JSON-Struktur einer READ-Response an /v1/erreichbarkeiten/{erreichbarkeitbeziehung.id}	162
Quellcode 92: Beispiel Response zu /v1/erreichbarkeiten/{erreichbarkeit.id} auf HTTP-GET	163
Quellcode 93: JSON-Struktur eines DELETE-Requests an /v1/erreichbarkeiten/{erreichbarkeit.id}	163
Quellcode 94: Beispiel Request zu /v1/erreichbarkeiten/{erreichbarkeit.id} auf HTTP-DELETE	163
<i>Quellcode 95: URL-Struktur des Endpunkts organisationen</i>	164
Quellcode 96: JSON-Struktur einer READ-Response an /v1/organisationen	166
Quellcode 97: Beispiel Response zu /organisationen auf HTTP-GET	166
Quellcode 98: Beispiel Response zu /organisationen auf HTTP-GET mit den Filter-parametern „?kennung=NI_12-34-56&typ=SCHULE“	167
<i>Quellcode 99: URL-Struktur des Endpunkts organisationen/{organisation.id}</i>	168
Quellcode 100: JSON-Struktur einer READ-Response an /v1/organisationen/{id}	169
Quellcode 101: Beispiel Response zu /organisationen/{organisation.id} auf HTTP-GET	170
<i>Quellcode 102: URL-Struktur des Endpunkts organisationen/{id}/organisationsbeziehungen</i>	170
Quellcode 103: JSON-Struktur einer READ-Response an /v1/organisationen/{id}/organisationsbeziehungen	172
Quellcode 104: Beispiel Response zu /v1/organisationen/{id}/organisationsbeziehungen auf HTTP-GET ohne Filter	173

Quellcode 105: Beispiel Response zu /v1/organisationen/{id}/organisationsbeziehungen auf HTTP-GET mit Filter "?hat_als_organisationsbeziehungen=nein&ist_von_organisationsbeziehungen=ja"	174
<i>Quellcode 106: URL-Struktur des Endpunkts organisation-info</i>	175
Quellcode 107: JSON-Struktur einer READ-Response an /v1/organisation-info	176
Quellcode 108: Beispiel Response zu /organisation-info auf HTTP-GET	177
Quellcode 109: URL-Struktur des Endpunkts Gruppen	178
Quellcode 110: JSON-Struktur eines CREATE-Requests an /v1/gruppen	180
Quellcode 111: JSON-Struktur einer CREATE-Response an /v1/gruppen	180
Quellcode 112: Beispiel Response zu /gruppen auf HTTP-POST	181
Quellcode 113: JSON-Struktur eines READ-Requests an /v1/personen	184
Quellcode 114: JSON-Struktur einer READ-Response an /v1/Gruppe	185
Quellcode 115: Beispiel Response zu /gruppen auf HTTP-GET	188
<i>Quellcode 116: URL-Struktur des Endpunkts gruppen/{gruppe.id}</i>	189
Quellcode 117: JSON-Struktur der Antwort bei READ auf /gruppen/{gruppe.id }	190
Quellcode 118: Beispiel Response zu /gruppen/{gruppe.id} auf HTTP-GET	191
Quellcode 119: JSON-Struktur eines UPDATE-Requests an /v1/gruppen/{gruppen.id}	195
Quellcode 120: JSON-Struktur einer UPDATE-Response an /v1/gruppen/{gruppe.id}	196
Quellcode 121: Beispiel Response zu /gruppen/{gruppe.id } auf HTTP-PUT	197
Quellcode 122: JSON-Struktur eines DELETE-Requests an /v1/gruppen/{gruppe.id}	198
Quellcode 123: Beispiel Request zu /v1/gruppen/{gruppe.id} auf HTTP-DELETE	198
<i>Quellcode 124: URL-Struktur des Endpunkts gruppen/{gruppe.id}/gruppenzugehoerigkeiten</i>	199
Quellcode 125: JSON-Struktur eines CREATE-Requests an /v1/gruppen/{gruppe.id }/gruppenzugehoerigkeiten	201
Quellcode 126: Beispiel Request zu /v1/gruppen/{gruppe.id }/ gruppenzugehoerigkeiten auf HTTP-POST	201
Quellcode 127: JSON-Struktur einer CREATE-Response an /v1/gruppen/{gruppe.id}/gruppenzugehoerigkeiten	202
Quellcode 128: Beispiel Response zu /v1/gruppen/{gruppe.id}/gruppenzugehoerigkeiten auf HTTP-POST	202
Quellcode 129: JSON-Struktur einer READ-Response an /v1/ gruppen/{gruppe.id}/gruppenzugehoerigkeiten	204
Quellcode 130: Beispiel Response zu /gruppen/{gruppe.id}/gruppenzugehoerigkeiten auf HTTP-GET	204



Quellcode 131: URL-Struktur des Endpunkts gruppenzugehoerigkeiten	205
Quellcode 132: JSON-Struktur einer READ-Response an /v1/gruppenzugehoerigkeiten	207
Quellcode 133: Beispiel Response zu /gruppenzugehoerigkeiten auf HTTP-GET	207
<i>Quellcode 134: URL-Struktur des Endpunkts gruppenzugehoerigkeiten/{gruppenzugehoerigkeit.id}</i>	209
Quellcode 135: JSON-Struktur einer READ-Response an /v1/gruppenzugehoerigkeiten/{gruppenzugehoerigkeit.id}	211
Quellcode 136: Beispiel Response zu /gruppenzugehoerigkeiten/{gruppenzugehoerigkeit.id} auf HTTP-GET	212
Quellcode 137: JSON-Struktur eines UPDATE-Requests an /v1/gruppenzugehoerigkeiten/{gruppenzugehoerigkeit.id}	214
Quellcode 138: JSON-Struktur einer UPDATE-Response an /v1/gruppenzugehoerigkeiten/{gruppenzugehoerigkeit.id}	215
Quellcode 139: Beispiel Response zu /gruppenzugehoerigkeiten/{gruppenzugehoerigkeit.id } auf HTTP-PUT	215
Quellcode 140: JSON-Struktur eines DELETE-Requests an /v1/gruppenzugehoerigkeiten/{gruppenzugehoerigkeit.id}	216
Quellcode 141: Beispiel Request zu /v1/gruppenzugehoerigkeiten/{gruppenzugehoerigkeit.id} auf HTTP-DELETE	216
Quellcode 142: URL-Struktur des Endpunkts codelisten	217
Quellcode 143: JSON-Struktur eines READ-Requests an /v1/codelisten	218
Quellcode 144: URL-Struktur des Endpunkts codelisten/{codeliste}	218
Quellcode 145: JSON-Struktur eines READ-Requests an /v1/codelisten/geschlecht	218
Quellcode 146: URL-Struktur des Endpunkts versionen	219
Quellcode 147: JSON-Struktur eines READ-Requests an /v1/versionen	220

## 1 Stand der Bearbeitung

Dieses Dokument befindet sich aktuell im Bearbeitungsstand „Veröffentlichte Spezifikation“ in Version 1.003.

**Achtung:** Die in diesem Dokument referenzierten Codelisten stellen nur eine erste Version dar und sind zum aktuellen Zeitpunkt nicht vollständig!

## 2 Vorbemerkung

Dieses Dokument beschreibt die Schnittstellen für SchulConneX. Diese Schnittstellenspezifikation wird iterativ erweitert. Dabei wird angestrebt, dass die Schnittstellen-Versionen mit der gleichen führenden Versionsnummer (Ziffer vor dem ersten Punkt) insofern kompatibel sind, dass auf einer älteren Version aufbauende Dienste auch mit neueren Versionen der Schnittstelle nutzbar bleiben.

## 3 Einleitung

Dieses Dokument spezifiziert den Schnittstellenstandard SchulConneX. SchulConneX dient dem Austausch und der Synchronisation von Identitäten und Kontextdaten zwischen schulischen Identitätsmanagementsystemen einerseits und Anwendungen andererseits. Die Definition und Bereitstellung dieser Schnittstellen (APIs) wird als Teil der im Rahmen der Kultusministerkonferenz „Bildung in der digitalen Welt“ im Jahre 2016 erstellten Strategie zur Einrichtung landesweiter ID-Management-Systeme im schulischen Bereich verstanden. Ziel ist, ein standardisiertes Anmeldeverfahren für den Zugang zu digitalen Bildungsplattformen und -medien zu schaffen.

Dabei ist herauszuheben, dass hiermit weder eine Vereinheitlichung der bei den Schulverwaltungen eingesetzten Systeme noch eine zentrale Datenhaltung beabsichtigt ist.

## 4 Allgemeines

### 4.1 Kernanwendungsfälle

Prinzipiell unterstützt die API zwei grundlegend unterschiedliche Anwendungsfälle: Zum einen die Bereitstellung von Nutzer- und Kontextdaten im Zusammenhang mit einem Single Sign-on für Anwendungen (Dienste), zum anderen die Synchronisation von Nutzer- und Kontextdaten zwischen Schulverwaltungssystemen (Quellsystemen).

Ein SchulConneX-Service ist ein Dienst, der den Vorgaben der SchulConneX-Spezifikation folgt. Der Zugriff auf alle Endpunkte der REST-APIs eines SchulConneX-Services wird durch einen zentralen Authentifizierungs- und Autorisierungs-Server gesichert. Ein Zugriff auf die Endpunkte darf nur mit einem gültigen, von diesem Server ausgestellten Access-Token möglich sein. Um Access-Token für die API-Nutzung zu erhalten, müssen Quellsysteme und Dienste als Clients mit einer Client-ID registriert sein.

### 4.2 Nutzung der Schnittstelle durch Dienste

Dienste (Anwendungen) nutzen die SchulConneX-Dienste-API zum Abruf von Nutzer- und Kontextdaten. Ein Zugriff auf das API ist in unterschiedlichen Sicherheitskontexten möglich:

Aufruf im **Sicherheitskontext einer angemeldeten nutzenden Person mit Zugriff auf den Personenkontext**, mit dem die Person angemeldet ist. Bei dieser Art des Zugriffs werden nur die Daten, die sich auf den konkreten Personenkontext beziehen, mit dem die oder der Nutzende angemeldet ist, durch die REST-API bereitgestellt. Dies ist der Standard-Zugriff für die Dienste. Dieser Zugriff setzt einen Authentifizierungs- und Autorisierungsvorgang des oder der Nutzenden durch den Server voraus, bei dem ein entsprechender Access-Token für den Zugriff auf den Endpunkt ausgestellt wird.

Aufruf im **Sicherheitskontext einer angemeldeten Person mit Zugriff auf alle Personenkontexte der Person**. Nur denen mit besonderem Vertrauensstatus kann es ermöglicht werden, auf die vollständigen Daten einer Person (alle Personenkontexte) zuzugreifen. Dieser Zugriff setzt ebenfalls einen Authentifizierungs- und Autorisierungsvorgang des oder der Nutzenden durch den Server voraus, bei dem ein entsprechender Access-Token für den Zugriff auf den Endpunkt ausgestellt wird.

Aufruf im **Sicherheitskontext des Dienstes mit Zugriff auf alle Daten**, die für sie freigegeben sind. Diese Art des Zugriffs erlaubt den Diensten eine (periodische) Synchronisierung der bei ihnen gehaltenen Daten mit den Daten des SchulConneX-Services, ohne dass sich die nutzenden Personen aktiv anmelden müssen. Der Zugriff erfolgt mit einem Access-Token, den der Authentifizierungs- und Autorisierungs-Server für den Dienst ausstellt. Dieser Sicherheitskontext wird aktuell zum periodischen Abgleich von gelöschten Daten verwendet.

### 4.3 Ablauf Löschen von Personenkontexten und Personen

Aus Sicht des Datenschutzes ist der SchulConneX-Server ein Datenübermittler zwischen Schulverwaltung (Quellsystemen) und angebundenen Anwendungen (Dienste). Aus dieser Voraussetzung leiten sich bestimmte gesetzliche Anforderungen an das Löschen von

Personen und Personenkontexten ab. Diese Anforderungen werden durch festgelegte Statusfolgen bei der Löschung konzeptuell umgesetzt.

Besonders relevant aus Sicht des Datenschutzgesetzes sind hierbei zwei Aspekte: Erstens die Unverzüglichkeit der Löschung und zweitens die Benachrichtigung anderer Organisationen, denen die Daten bereitgestellt wurden, über die Löschung der Daten. Diese Organisationen sind dann aufgefordert, eventuelle Kopien der Daten ihrerseits unverzüglich zu löschen.

Der SchulConneX-Service macht vom Ablauf her keinen Unterschied zwischen der Löschung von Personenkontexten und Personen aus organisatorischen Gründen (wie Schuljahr endet, Lehrkraft verlässt die Schule) und der Löschung aufgrund von Art. 17 DSGVO ("Recht auf Löschung" / "Recht auf Vergessenwerden"). Im letzteren Fall ist jedoch die Frist zur Löschung deutlich kürzer, technisch wird meist der aktuelle Tag als Löschtermin gesetzt.

Primäre Quellsysteme, synchronisierte Quellsysteme (über Sichtfreigaben) und Dienste können den Löschtermin nutzen, um Nutzer und Nutzerinnen über eine bevorstehende Löschung zu informieren und eventuell zu Aktionen (wie eigene Sicherung der Daten) aufzufordern.

#### 4.3.1 Löschen von Personenkontexten und Personen aus Sicht des primären Quellsystems

Um einen Personendatensatz und die dazugehörigen Personenkontexte zu löschen, geht das Quellsystem wie folgt vor:

1. Löschen aller Personenkontexte zu einer Person
2. Löschen der Personen

Allgemein werden Personenkontexte durch das Setzen eines Löschzeitpunktes gelöscht. Wurden Personenkontexte noch nicht von angebundenen Diensten genutzt, so können diese auch direkt mit dem HTTP DELETE Aufruf der API "personenkontexte/{id}" gelöscht werden. Wurden die Daten jedoch bereits an Dienste weitergeleitet, ist die direkte Löschung nicht mehr möglich und die Löschung muss durch Setzen des Löschzeitpunktes ausgelöst werden.

Dabei ist der Vorgang wie folgt:

- Es wird ein Löschzeitpunkt für jeden Personenkontext der Person gesetzt.
- Dieser Löschzeitpunkt darf nicht in der Vergangenheit liegen.
- Zum Löschzeitpunkt wird der Personenkontext vom SchulConneX-Service gelöscht, die darin vorhandenen Daten sind nicht mehr direkt zugreifbar.
- Im Fall, dass es sich um den einzigen Personenkontext der Person handelt, kann nach der Löschung dieses Personenkontexts auch die Person mit einem DELETE an "personen/{id}" gelöscht werden.

#### 4.3.2 Information von Diensten über die Löschung von Personenkontexten

Die DSGVO schreibt vor, dass Dienste, welchen personenbezogene Daten bereitgestellt wurden, über die Löschung der Daten zu informieren sind, um diesen zu ermöglichen, alle Links zu diesen personenbezogenen Daten oder Kopien oder Replikationen dieser personenbezogenen Daten unverzüglich zu löschen.

Um dies zu ermöglichen, wird für Dienste, welche persistent Daten speichern, der API-Endpunkt `"/personen-info"` bereitgestellt.

Diese Schnittstelle liefert zu allen aktiven Personenkontexten, welche an das System ausgeliefert wurden und die nicht gelöscht sind, die (pseudonymisierte) Kontext-ID sowie, soweit gesetzt, den geplanten Löschozeitpunkt. Letzterer ist nur als Absichtserklärung zu verstehen und kann dazu dienen, die Nutzerin oder den Nutzer darauf hinzuweisen, dass der Zugang demnächst endet, und eventuell aufzufordern, relevante Daten vor dem Termin zu sichern. Die Kenntnis des Löschozeitpunkts allein ist aber nicht ausreichend, um die Löschung vorzunehmen, da der Löschozeitpunkt gelöscht oder geändert werden kann.

Es ist Aufgabe des Dienstes, den API-Endpunkt `"/personen-info"` mit ausreichender Regelmäßigkeit abzurufen, damit eine angemessen schnelle Löschung erfolgen kann.

Hat ein Dienst zu einer Person alle lokal persistent gehaltenen Personenkontext-Daten gelöscht, so muss er auch die persistent gehaltenen Daten zur Person löschen.

## 5 Datenmodell der Schnittstelle

Der Begriff Datenmodell steht im Kontext dieser Spezifikationen nicht für das Datenmodell zur Persistierung von Entitäten in einer Datenhaltung. Im Kontext dieser Spezifikation repräsentiert es die Datenstruktur der Nutzlast von Anfragen (Request Payload) und Nutzlast der Antworten (Response Payload) bei XHR-Anfragen (HTTP-Anfragen gegen API-Endpunkte).

Das Datenmodell der Schnittstelle basiert auf Datenmodellen der Spezifikationen XBildung und XSchule (bspw. „NatuerlichePerson“ ist die Vorlage für „Person“, auch „Organisation“ ist an XBildung orientiert), jeweils mit spezifischen Erweiterungen.

In den Datenmodellen wird für jedes Attribut die Anzahl des Vorkommens definiert. Attribute mit der Anzahl 0..1 oder 0..n sind als optional zu verstehen. Je nach Autorisierung kann bei einer erfolgreichen HTTP-Anfrage gegen einen API-Endpunkt das minimale Datenmodell ausgegeben werden. Dieses umfasst die Attribute mit der Anzahl 1. Beispielsweise kann eine Rückgabe nur das Attribut `id` (identifiziert) enthalten.

Bei Attributen vom Typ String (Zeichenkette) ist die maximale Länge der Zeichenkette standardmäßig mit 256 Zeichen definiert. Bei Strings, deren maximale Länge davon abweicht, ist die maximale Länge in den Datenmodellen angegeben. Diese Angabe erfolgt durch Anhang der Maximallänge in Klammern hinter den Datentyp String. Die Angabe „String (32)“ definiert einen Text mit der maximalen Länge von 32 Zeichen.

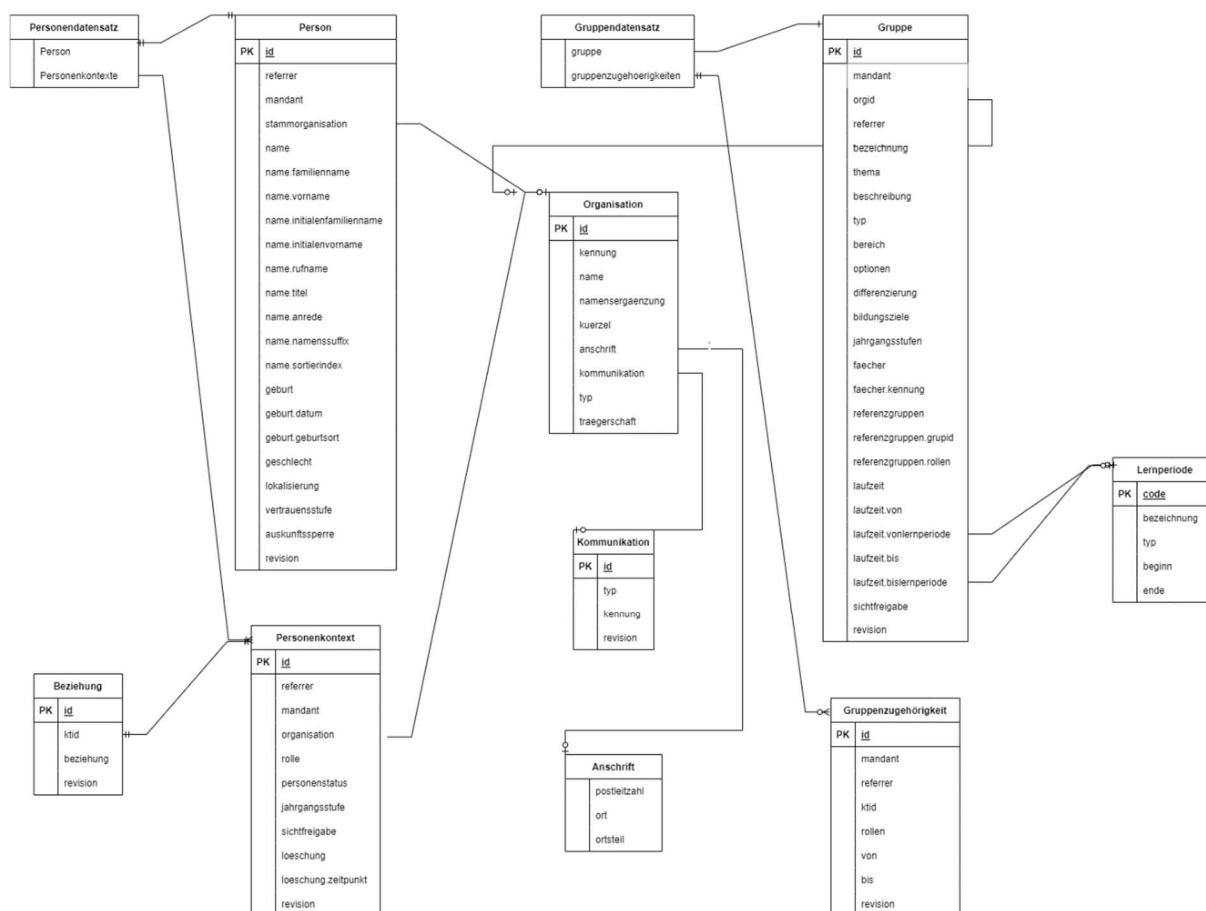


Abbildung 1: Graphische Darstellung des Datenmodells

## 5.1 Datumsformat

Alle Datumsangaben werden, entsprechend der ISO-8601-Formatierung **YYYY-MM-DD** ausschließlich als vierstellige Jahreszahl, gefolgt von der zweistelligen Monatsnummer, gefolgt vom zweistelligen Tag innerhalb des Monats, jeweils getrennt durch Bindestriche dargestellt. Negative Jahreszahlen sind nicht zulässig, ebenso wenig wie Jahreszahlen ohne Jahrhundertangabe, Monatsnamen oder einstellige Monats- und Tagesangaben. Somit ist 2022-08-19 eine zulässige Datumsangabe. 22-08-19, 2022-8-19, 2022-AUG-19 oder 2022-AU-19 sind keine zulässigen Datumsdarstellungen.

Einen Sonderfall stellt hier das Format zur Darstellung eines Löschezitpunktes dar. Anders als bei sonstigen Datumsangaben, welche nur tagesgenau benötigt werden, ist für den Löschezitpunkt auch die Uhrzeit relevant. Das Format für einen Löschezitpunkt ist **yyyy-MM-dd'T'hh:mm'Z'** als UTC-Zeitpunkt.

## 5.2 Organisation

Eine Organisation stellt eine Schule oder sonstige relevante Organisation im Kontext EINES SchulConneX-Services dar. Personen werden diesen Organisationen über ihre Personenkontexte mit entsprechenden Rollen zugeordnet. In der aktuellen Version der

Schnittstellendefinition ist für Datensätze des Datentyps Organisation die Erstellung, Löschung und Modifikation nur durch einen organisatorischen Prozess außerhalb der API vorgesehen. Das Datenmodell Organisation wird in einer zukünftigen Version der Schnittstellendefinition das Attribut "revision" führen; dies wird mit der Einführung der Funktionalität zum Aktualisieren und Löschen eines Datensatzes des Datentyps Organisation notwendig sein. Diesbezüglich wird in dem aktuellen Datenmodell das Attribut "revision" noch nicht geführt.

Attribut	Typ	Anz.	Bemerkung
<b>id</b>	String (UTF-8)	1	ID der Organisation. Wird vom SchulConneX-Server vergeben und ist eindeutig. Dieses Attribut ist unveränderbar (immutable).
<b>kennung</b>	String (UTF-8)	0..1	Die optionale Kennung (Identifikations-ID) einer "Organisation" muss innerhalb eines Organisationstyps eindeutig sein.
<b>name</b>	String (DIN 91379.B)	0..1	offizieller Name einer Organisation
<b>namensergaenzung</b>	String (DIN 91379.B)	0..1	ergänzender Name einer Organisation
<b>kuerzel</b>	String (64) (DIN 91379.B)	0..1	Kurzname einer Organisation, maximal 64 Zeichen
<b>anschrift</b>	Anschrift	0..1	Anschrift der Organisation
<b>kommunikation</b>	Kommunikation[]	0..1	Array von Kommunikations-Daten (siehe Datenmodell 5.10 Kommunikation) über welche die Organisation erreichbar ist
<b>typ</b>	String (Code)	0..1	Typ der Organisation, Referenz auf einen Code der Codeliste 12.5 Organisationstyp
<b>traegerschaft</b>	String (Code)	0..1	Art der Trägerschaft der Organisation, Referenz auf einen Code der Codeliste 12.8 Trägerschaft

### Beispiel JSON des Datentyps Organisation

```
{
  "id": "b0d7b0dd-3477-4122-a38d-095ec242e786",
  "kennung": "NI_12345",
  "name": "Heinrich-Heine-Gymnasium",
  "namensergaenzung": "vorher Heinrich-Heine-Gesamtschule",
  "kuerzel": "Heine-Gym",
  "typ": "SCHULE",
  "traegerschaft": "02"
}
```

Quellcode 1: Beispiel JSON für eine Organisation

## 5.3 Person

Nachfolgend ist das Datenmodell einer Person dargestellt.

Attribut	Typ	Anz.	Bemerkung
<b>id</b>	String (UTF-8)	1	ID der Person. Wird vom SchulConneX-Server vergeben, ist eindeutig. Dieses Attribut ist unveränderbar (immutable).
<b>referrer</b>	String (UTF-8)	0..1	Die optionale Identifikations-ID einer Person muss innerhalb eines Quellsystems eindeutig sein (z. B. die einheitliche Personalnummer des Landes Niedersachsen, falls vergeben).



Attribut	Typ	Anz.	Bemerkung
<b>mandant</b>	String (UTF-8)	1	ID des Mandanten, dem die Personen zugeordnet ist. Wird vom SchulConneX-Server vergeben und ist eindeutig.
<b>stammorganisation</b>	String (UTF-8)	0..1	Personen können einer Organisation angehören, jedoch zeitweise an einer anderen Organisation tätig sein.
<b>name</b>	<i>Strukturierung</i>	1	
<b>name.familienname</b>	String (DIN 91379.A)	1	Familienname(n) der Person  Mehrere Familiennamen werden durch Leerzeichen separiert.
<b>name.vorname</b>	String (DIN 91379.A)	1	Mehrere Vornamen werden durch Leerzeichen separiert.
<b>name.initialenfamilienname</b>	String(8) (DIN 91379.A)	0..1	Initial oder Initialen des Familiennamens, maximal acht Zeichen
<b>name.initialenvorname</b>	String(8) (DIN 91379.A)	0..1	Initial oder Initialen des Vornamens, maximal acht Zeichen
<b>name.rufname</b>	String(32) (DIN 91379.A)	0..1	Bei mehreren Vornamen wird der täglich im

Attribut	Typ	Anz.	Bemerkung
			Gebrauch genutzte Vorname geführt, maximal 32 Zeichen.
<b>name.titel</b>	String (DIN 91379.B)	0..1	Titel der Person (bspw. ein akademischer Grad, Dienst- und Amtsbezeichnungen oder militärische Ränge)
<b>name.anrede</b>	String (64/512) (DIN 91379.B)	0..n	Liste (Array) von Namenszusätzen bei der Anrede  Jeder einzelne Namenszusatz hat eine Maximallänge von 64 Zeichen, die Maximallänge für alle Namenszusätze liegt bei 512 Zeichen.
<b>name.namenssuffix</b>	String (64/1024) (DIN 91379.A)	0..n	Liste (Array) von Namenssuffixen. Jeder einzelne Namenssuffix hat eine Maximallänge von 64 Zeichen, die Maximallänge für alle Namenssuffixe liegt bei 1024 Zeichen.
<b>name.sortierindex</b>	String (Nummer)	0..1	Index des Buchstabens des Nachnamens, nach dem sortiert werden soll. Der Index ist Null-basiert, d. h. ein Wert von "4" zeigt an, dass die ersten vier Buchstaben ignoriert und ab dem

Attribut	Typ	Anz.	Bemerkung
			fünften Buchstaben des Nachnamens sortiert werden soll. Ist kein Sortierindex angegeben, so ist der Nachname entsprechend dem ersten Buchstaben in Listen einzusortieren, äquivalent zum Sortierindex "0".
<b>geburt</b>	<i>Strukturierung</i>	0..1	
<b>geburt.datum</b>	Datum (siehe Abschnitt 5.10)	0..1	Geburtsdatum einer Person
<b>geburt.geburtsort</b>	String (DIN 91379.A)	0..1	Geburtsort einer Person, bestehend aus Ortsname und Land, beispielsweise "Eindhoven, Niederlande"  Wird kein Land angegeben, so wird "Deutschland" angenommen.
<b>geschlecht</b>	String (Code)	0..1	Referenz auf einen Code der Codeliste 12.2 Geschlecht
<b>lokalisierung</b>	String (Code)	0..1	bevorzugte Lokalisierungseinstellung einer Person für Anwendungen gemäß RFC5646  Wenn nicht angegeben, wird „de“ angenommen.

Attribut	Typ	Anz.	Bemerkung
			Referenz auf einen Code der Codeliste 12.9 Lokalisierung.
<b>vertrauensstufe</b>	String (Code)	0..1	gibt an, wie stark die Personendaten vom erfassenden Mandanten verifiziert wurden,  Referenz auf einen Code der Codeliste 12.4 Vertrauensstufe
<b>auskunftssperre</b>	String (Code)	1	Auskunftssperre über eine Person  Dienste erhalten nur einen uneingeschränkten Datensatz, wenn der Wert "NEIN" ist. Sofern nicht explizit "JA" eingegeben wurde, ist der Default "NEIN".  Referenz auf einen Code der Codeliste 12.7  Boolean
<b>revision</b>	String (UTF-8)	1	Revision der Person;  Wird vom SchulConneX - Server mit der Erstellung des Datensatzes sowie Aktualisierung generiert  Dieser Wert kann nicht von Quellsystemen oder Diensten gesetzt werden.

## *Datenmodell 2: Datenmodell einer Person*

## Beispiel JSON des Datentyps Person

```
{
  "id": "a6e1a860-8d44-4b2b-aef7-aa2c8bf5beb5",
  "referrer": "125",
  "mandant": "58f45270-8e54-40c6-a212-980307fc19be",
  "name": {
    "familienname": "von Musterfrau",
    "vorname": "Natalie Lisa",
    "initialenfamilienname": "M.",
    "initialenvorname": "N.",
    "rufname": "Natalie",
    "titel": "Dr.",
    "anrede": [
      "Frau"
    ],
    "namenssuffix": [
      "jun."
    ],
    "sortierindex": "4"
  },
  "geburt": {
    "datum": "2005-05-01",
    "geburtsort": "Berlin, Deutschland"
  },
  "geschlecht": "w",
  "lokalisierung": "de-DE",
  "vertrauensstufe": "VOLL",
  "auskunftssperre": "NEIN",
  "revision": "1"
}
```

*Quellcode 2: Beispiel JSON einer Person*

## 5.4 Personenkontext

Der Personenkontext gibt an, in welcher Rolle der Dienst von nutzenden Personen in Anspruch genommen wird. Der Personenkontext umfasst neben der Rolle auch Informationen über die Organisation, an der diese Rolle ausgeführt wird.

Eine Person kann einen oder mehrere Personenkontexte besitzen, indem sie beispielsweise an derselben Schule sowohl die Rollen „Schulleiter“ als auch „Erziehungsberechtigter“ hat, oder an einer Schule die Rolle „Lehrender“ und an einer weiteren Organisation „Lernender“ ist.

Es dürfen jedoch keine inhaltlich identischen Personenkontexte angelegt werden. Insbesondere ist es unzulässig, dieselbe Kombination aus Organisation und Rolle einer Person in unterschiedlichen Personenkontexten zuzuweisen.

Bei der Abfrage einer Person über die API können mehrere zur Person gehörenden Personenkontexte als Liste mit zurückgegeben werden.

Nicht alle Kombinationen von Attributwerten sind sinnvoll. So ist beispielsweise die Kombination einer Jahrgangsstufe mit einer anderen Rolle als „LERN“ in der Regel nicht sinnvoll.

Attribut	Typ	Anz.	Bemerkung
id	String (UTF-8)	1	ID des Personenkontexts Wird vom SchulConneX-Server vergeben und ist eindeutig. Dieses Attribut ist unveränderbar (immutable).
referrer	String (UTF-8)	0..1	ID des Personenkontexts im Quellsystem Wird vom Quellsystem vergeben und muss im Quellsystem eindeutig sein.
mandant	String (UTF-8)	1	ID des Mandanten, dem der Personenkontext zugeordnet ist

			Wird vom SchulConneX-Server vergeben und ist eindeutig.
organisation	String (Code)	1	Organisation. (Siehe 5.2 Datenmodell Organisation)
rolle	String (Code)	1	Rolle der Person innerhalb der Organisation, Referenz auf einen Code der Codeliste 12.3 Rolle Rolle.
personenstatus	String (Code)	0..1	Status, den eine Person in einer Organisation in Bezug auf eine bestimmte Rolle hat,Referenz auf einen Code der Codeliste 12.1 Personenstatus
jahrgangsstufe	String (Code)	0..1	Jahrgangsstufe, die eine Person in der Organisation in einer bestimmten Rolle besucht,Referenz auf Code der Codeliste 12.6 Jahrgangsstufe
sichtfreigabe	String (Code)	0..1	gibt an, ob dieser Personenkontext aufgrund der Freigabe durch eine andere Organisation sichtbar ist  Ist "sichtfreigabe" nicht gesetzt, so entspricht das dem Wert "nein". Der Wert von Sichtfreigabe ist Boolean nach Codetabelle 12.7 Boolean.



loeschung	Struktur	0..1	Aktuell gibt es zu Löschungen nur ein Attribut (Zeitpunkt), eventuell werden in späteren Versionen weitere Attribute hinzugefügt.
loeschung.zeitpunkt	String (datetime)	1	Datum und Uhrzeit der Löschung des Personenkontexts  Das Format des Löschezitpunktes ist yyyy-MM-dd'T'hh:mm'Z' als UTC-Zeitpunkt. Solange ein Personenkontext nicht gelöscht wurde, kann das Attribut loeschzeitpunkt geändert oder gelöscht werden. Ist das Attribut loeschung vorhanden, so muss der Zeitpunkt gesetzt sein.
revision	String (UTF-8)	1	Revision des Personenkontext  wird vom SchulConneX-Server mit der Erstellung des Datensatzes sowie Aktualisierung generiert  Dieser Wert kann nicht von Quellsystemen oder Diensten gesetzt werden.

Datenmodell 3: Datenmodell eines Personenkontexts

## Beispiel JSON des Datentyps Personenkontext

```
{
  "id": "4d0f579c-0b9a-4d3a-b484-87b3bee8a2ad",
  "referrer": "NI_12345_12554648",
  "mandant": "58f45270-8e54-40c6-a212-980307fc19be",
  "organisation": {
    "id": "b0d7b0dd-3477-4122-a38d-095ec242e786",
    "kennung": "NI_12345",
    "name": "Heinrich-Heine-Gymnasium",
    "namensergaenzung": "vorher Heinrich-Heine-Gesamtschule",
    "kuerzel": "Heine-Gym",
    "anschrift" : {
      "postleitzahl": "30519",
      "ort": "Hannover",
      "ortsteil": "Döhren"
    },
    "typ": "SCHULE",
    "traegerschaft": "02"
  },
  "rolle": "LERN",
  "personenstatus": "AKTIV",
  "jahrgangsstufe": "05",
  "revision": "1"
}
```

Quellcode 3: Beispiel JSON eines Personenkontexts

## 5.5 Personendatensatz

Datenmodell eines Personendatensatzes: Personendatensatz ist als Daten-Container (WrapperObject) zu verstehen. Über Personendatensatz werden Person (Attribut "personendatensatz.person") sowie die Personenkontexte (Attribut "personendatensatz.personenkontexte") zusammengeführt.

Attribut	Typ	Anz.	Bemerkung
----------	-----	------	-----------

<b>person</b>	Person	1	Personen. (siehe 5.3 Datenmodell Personen)
<b>personenkontexte</b>	Personenkontext	0..n	Liste (Array) von Personenkontexten. (siehe 5.4 Datenmodell Personenkontext)

Datenmodell 4: Datenmodell des Personendatensatzes

### Beispiel JSON des Datentyps Personendatensatz

```
{
  "person": {
    "id": "a6e1a860-8d44-4b2b-aef7-aa2c8bf5beb5",
    "referrer": "125",
    "mandant": "58f45270-8e54-40c6-a212-980307fc19be",
    "name": {
      "familienname": "von Musterfrau",
      "vorname": "Natalie Lisa",
      "initialenfamilienname": "M.",
      "initialenvorname": "N.",
      "rufname": "Natalie",
      "titel": "Dr.",
      "anrede": [
        "Frau"
      ],
      "namenssuffix": [
        "jun."
      ],
      "sortierindex": "4"
    },
    "geburt": {
      "datum": "2005-05-01",
      "geburtsort": "Berlin, Deutschland"
    },
    "geschlecht": "w",
    "lokalisierung": "de-de",
  }
}
```

```

    "vertrauensstufe": "VOLL",
    "auskunftssperre": "NEIN",
    "revision": "1"
  },
  "personenkontexte": [
    {
      "id": "4d0f579c-0b9a-4d3a-b484-87b3bee8a2ad",
      "referrer": "NI_12345_123",
      "mandant": "58f45270-8e54-40c6-a212-980307fc19be",
      "organisation": {
        "id": "b0d7b0dd-3477-4122-a38d-095ec242e786",
        "kennung": "NI_12345",
        "name": "Heinrich-Heine-Gymnasium",
        "namensergaenzung": "vorher Heinrich-Heine-Gesamtschule",
        "kuerzel": "Heine-Gym",
        "anschrift" : {
          "postleitzahl": "30519",
          "ort": "Hannover",
          "ortsteil": "Döhren"
        },
        "typ": "SCHULE",
        "traegerschaft": "02"
      },
      "rolle": "LERN",
      "personenstatus": "AKTIV",
      "jahrgangsstufe": "01",
      "revision": "1"
    }
  ]
}

```

Quellcode 4: Beispiel JSON eines Personendatensatzes

## 5.6 Gruppe

Eine Gruppe erlaubt es, mehrere Personen in einer Gruppe zusammenzufassen. Typischerweise handelt es sich bei einer Gruppe um eine Schulklasse oder eine ähnliche Struktur in einer Schule oder anderen Organisation. Die Zugehörigkeit einer Person zu einer Gruppe wird durch das Datenmodell Gruppenzugehörigkeit beziehungsweise zusammen mit der Gruppen-Information im Gruppendatensatz abgebildet.

Attribut	Typ	Anz.	Bemerkung
id	String (UTF-8)	1	ID der Gruppe wird vom SchulConneX-Server vergeben und ist eindeutig Dieses Attribut ist unveränderbar (immutable).
mandant	String (UTF-8)	1	ID des Mandanten, dem die Gruppe zugeordnet ist wird vom SchulConneX-Server vergeben und ist eindeutig Dieses Attribut ist unveränderbar (immutable).
orgid	String (UTF-8)	0..1	ID der Organisation, welcher die Gruppe zugeordnet ist oft identisch mit Mandanten, kann aber in Einzelfällen davon abweichen
referrer	String (UTF-8)	0..1	ID der Gruppe im Quellsystem wird vom Quellsystem vergeben und muss im Quellsystem eindeutig sein
bezeichnung	String (UTF-8)	1	beschreibt die Bezeichnung der Gruppe, beispielsweise „Englisch, 2. Klasse“
thema	String (UTF-8)	0..1	Thema der Gruppe in Kurzform, idealerweise <35 Zeichen

beschreibung	String (1024) (UTF-8)	0..1	zusätzliche optionale Beschreibung der Gruppe
typ	String (Code)	1	Typ der Gruppe (Klasse, Kurs)  Referenz auf Code der Codeliste 12.13 Gruppentyp.
bereich	String (Code)	0..1	Art der Gruppe (Pflicht, Wahl, Grundkurs, Leistungskurs...)  Referenz auf Code der Codeliste 12.10 Gruppenbereich.
optionen	String (Code)	0..n	zusätzliche Optionen zur Gruppe, meistens sprachbezogen (bilingual, herkunftssprachlich)  Referenz auf Code der Codeliste 12.12 Gruppenoption
differenzierung	String (Code)	0..1	Differenzierung bei integriertem Unterricht oder in der gymnasialen Oberstufe (Erweiterungsebene, Grundebene)  Referenz auf Code der Codeliste 12.11 Gruppendifferenzierung
bildungsziele	String (Code)	0..n	Bildungsziel (Realschule, Hauptschule, Gymnasium, ...)  Referenz auf Code der Codeliste 12.18 Bildungsziel
jahrgangsstufen	String (Code)	0..n	Jahrgangsstufe (Jahrgangsstufe 1 bis Jahrgangsstufe 10)  Referenz auf Code der Codeliste 12.6 Jahrgangsstufe
faecher	Strukturierung	0..n	Liste der in der Gruppe behandelten Fächer

faecher.kennung	String (Code)	0..1	In der Gruppe behandelte Fächer  Referenz auf Code der Codeliste 12.17 Fächerkanon
referenzgruppen	Strukturierung	0..n	<p>Array von Referenzgruppen zu dieser Gruppe</p> <p>Bei Referenzgruppen handelt es sich um Gruppen, deren Mitglieder vollständig oder teilweise der Hauptgruppe zugeordnet werden.</p> <p>Referenzgruppen erlauben die einfachere Verwaltung von Gruppen von Personen, da hierdurch beispielsweise eine oder mehrere Klassen als Einheit einem Kurs als Teilnehmer zugeordnet werden können.</p> <p>Mitglieder von Referenzgruppen werden automatisch als Mitglieder der aktuellen Gruppe übernommen und behalten dabei ihre Rollen aus der Gruppenzugehörigkeit der Referenzgruppe. Die automatische Übernahme kann auf bestimmte Rollen beschränkt werden.</p> <p>Referenzgruppen müssen zur selben Organisation wie die Hauptgruppe gehören (Gruppe und Referenzgruppe müssen eine identische <code>orgid</code> haben).</p> <p>Referenzgruppen dürfen hierarchisch sein, d. h. Referenzgruppen von Referenzgruppen einer Gruppe werden behandelt wie Referenzgruppen der Gruppe. Zyklische Verweise sind jedoch untersagt und führen zu einer Fehlermeldung.</p>

referenzgruppen.grupid	String (UTF-8)	1	ID der Referenzgruppe
referenzgruppe.rollen	String (Code)	0..n	<p>Gruppenrollen aus der Gruppenzugehörigkeit von Personen, welche in die Referenzgruppe übernommen werden</p> <p><i>Hinweis:</i> Werden hier keine Gruppenrollen aufgeführt, so werden <i>alle</i> Personen der Referenzgruppe übernommen.</p>
laufzeit	<i>Strukturierung</i>	1	<p>Die Laufzeiten von Gruppen können entweder direkt durch Datumsangaben festgelegt werden (von/bis) oder durch Referenzen auf Lernperioden (vonlernperiode/bislernperiode). Wird eine Lernperiode referenziert, so steht "vonlernperiode" für das Anfangsdatum der Lernperiode (also das Attribut Beginn) und "bislernperiode" für das Enddatum der Lernperiode.</p> <p>Eine Mischung von Datum- und Perioden-Angaben ist nicht zulässig. Wird der Beginn der Laufzeit als "von" angegeben, so ist das Ende als "bis" anzugeben. Bei Nutzung von "vonlernperiode" ist das Ende mit "bislernperiode" zu spezifizieren.</p>
laufzeit.von	Datum (siehe Abschnitt 5.1)	0..1	<p>Anfang des Gültigkeitszeitraumes der Gruppe</p> <p>Dieser Zeitpunkt kann auch in der Zukunft liegen.</p>
laufzeit.vonlernperiode	String (Code)	0..1	Code der zu einer Gruppe gehörenden Lernperiode



			Referenz auf Code der Codeliste 12.15 Lernperiode
laufzeit.bis	Datum (siehe Abschnitt 5.1)	0..1	Ende des Gültigkeitszeitraumes der Gruppe
laufzeit.bislernperiode	String (Code)	0..1	Code der zu einer Gruppe gehörenden Lernperiode  Referenz auf Code der Codeliste 12.15 Lernperiode
sichtfreigabe	String (Code)	0..1	gibt an, ob diese Gruppe aufgrund der Freigabe durch eine andere Organisation sichtbar ist  Ist "sichtfreigabe" nicht gesetzt, so entspricht das dem Wert "nein". Der Wert von Sichtfreigabe ist Boolean nach Codetabelle 12.7 Boolean.
revision	String (UTF-8)	1	Revision der Gruppe  wird vom SchulConneX-Server mit der Erstellung des Datensatzes sowie Aktualisierung generiert  Dieser Wert kann nicht von Quellsystemen oder Diensten gesetzt werden.

Datenmodell 5: Datenmodell der Gruppe

## 5.7 Gruppenzugehörigkeit

Die Gruppenzugehörigkeit repräsentiert eine Verknüpfung zwischen einem Personenkontext und einer Gruppe. Hierin werden zusätzlich zu der Rolle im Personenkontext, welcher sich auf die Rolle innerhalb einer Organisation bezieht, gruppenspezifische Rollen angegeben. So kann beispielsweise jemand in einer Organisation die allgemeine Rolle "LERN" (Lernender oder Lernende) haben, in einer Gruppe zusätzlich noch Funktionen wie Klassensprecher oder Klassensprecherin ausüben.

Attribut	Typ	Anz.	Bemerkung
id	String (UTF-8)	1	ID der Gruppenzugehörigkeit
mandant	String (UTF-8)	1	ID des Mandanten, dem die Gruppenzugehörigkeit zugeordnet ist  wird vom SchulConneX-Server vergeben und ist eindeutig  Dieses Attribut ist unveränderbar (immutable).
referrer	String (UTF-8)	0..1	ID der Gruppenzugehörigkeit im Quellsystem  wird vom Quellsystem vergeben und muss im Quellsystem eindeutig sein
ktid	String (UTF-8)	1	ID des Personenkontexts, welchem die Gruppe zugeordnet ist
rollen	String (Code)	1..n	Rolle der Person innerhalb der Gruppe  Referenz auf Liste von Rollen der Codeliste 12.1412.14 Gruppenrolle
von	Datum (siehe Abschnitt 5.1)	0..1	Beginn der Gruppenzugehörigkeit  Dieser Zeitpunkt kann auch in der Zukunft liegen.
bis	Datum (siehe Abschnitt 5.1)	0..1	Ende der Gruppenzugehörigkeit
revision	String (UTF-8)	1	Revision der Gruppenzugehörigkeit

			<p>wird vom SchulConneX-Server mit der Erstellung des Datensatzes sowie Aktualisierung generiert</p> <p>Dieser Wert kann nicht von Quellsystemen oder Diensten gesetzt werden.</p>
--	--	--	--

Datenmodell 6: Datenmodell der Gruppenzugehörigkeit

## 5.8 Gruppendatensatz

Datenmodell eines Gruppendatensatzes: Der Gruppendatensatz ist als Daten-Container (WrapperObject) zu verstehen. Über den Gruppendatensatz werden Gruppe (Attribut "gruppendatensatz.gruppe") sowie die Gruppenzugehörigkeiten (Attribut "gruppendatensatz.gruppenzugehoerigkeit") zusammengeführt.

Attribut	Typ	Anz.	Bemerkung
<b>gruppe</b>	Gruppe	1	Gruppe. siehe 5.6 Gruppe)
<b>gruppenzugehoerigkeiten</b>	Gruppenzugehoerigkeit	0..n	Liste (Array) von Gruppenzugehörigkeiten. (siehe 5.7 Gruppenzugehörigkeit)

Datenmodell 7: Datenmodell des Gruppendatensatzes

### Beispiel JSON des Datentyps Gruppendatensatz

```
{
  "gruppe": {
    "id": "b3201d00-f21f-4986-a39d-02a09c8da26c",
    "mandant": "9b3f36ad-9d15-49f9-9660-6cf9746ba446",
    "orgid": "9b3f36ad-9d15-49f9-9660-6cf9746ba446",
    "referer": "HHG-1281b688-d639",
    "bezeichnung": "Englischkurs Klasse 3b",
    "typ": "KURS",
```

```

    "bereich": "PFLICHT",
    "optionen": [
        "01"
    ],
    "differenzierung": "G",
    "bildungsziele": [
        "RS"
    ],
    "jahrgangsstufen": [
        "03"
    ],
    "faecher": [{
        "kennung": "EN"
    }],
    "laufzeit": {
        "von": "2022-08-01",
        "bis": "2023-07-31"
    },
    "revision": "1"
},
"gruppenzugehoerigkeiten": [{
    "id": "8722b058-8747-4646-8da0-d523a9f619b8",
    "mandant": "cb4b7ee1-8651-40a8-b29d-b1bde7f6d21a",
    "referrer": "adf17dbe-7a72-45de-8c91-5b036fd080c8",
    "ktid": "42865c3f-2806-4488-9d46-dbaeb004bc8e",
    "rollen": ["LERN"],
    "von": "2022-08-01",
    "bis": "2023-07-31",
    "revision": "1"
},
{
    "id": " ed4a1432-6a58-4e99-b89e-e1d82b0052de",
    "mandant": "9b3f36ad-9d15-49f9-9660-6cf9746bb4559",
    "referrer": "adf17dbe-7a72-45de-8c91-5b036fd080c8",

```

```

        "ktid": "dc8bfbb0-a6ab-434a-a52c-bde200385d97",
        "rollen": ["LERN"],
        "von": "2022-08-01",
        "bis": "2023-07-31",
        "revision": "1"
    }
]
}

```

Quellcode 5: Beispiel JSON eines Gruppendatensatzes

## 5.9 Lernperiode

Eine Lernperiode beschreibt einen Zeitraum, in denen Gruppen von Organisationen angeboten werden können. Meist handelt es sich bei solchen Zeiträumen um gesetzlich vorgegebene Schuljahre oder Schulhalbjahre.

Beim Datentyp Lernperiode handelt es sich nicht um über API-Calls veränderbare Daten, sondern Lernperioden werden über eine Codeliste bereitgestellt. Anders als die meisten anderen Codelisten handelt es sich bei Lernperioden jedoch nicht um eine einfache Code-/Werte-Liste, sondern um einen komplexen Datentyp, dessen Struktur in der folgenden Tabelle beschrieben ist.

Attribut	Typ	Anz.	Bemerkung
code	String (UTF-8)	1	Code der Lernperiode
bezeichnung	String (UTF-8)	1	Beschreiben der Bezeichnung der Lernperiode, beispielsweise „Schuljahr 2023/2024“
typ	String (Code)	1	Typ der Lernperiode Referenz auf Liste von Typen der Codeliste 12.16 Lernperiodentyp

beginn	Datum (siehe Abschnitt 5.1)	1	Datum, an dem die Lernperiode beginnt
ende	Datum (siehe Abschnitt 5.1)	1	Datum an dem die Lernperiode endet

*Datenmodell 8: Datenmodell der Lernperiode*

## 5.10 Kommunikation

Das Datenmodell Kommunikation beschreibt, auf welchen Wegen Personen oder Organisationen erreicht werden können.

Attribut	Typ	Anz.	Bemerkung
id	String (UTF-8)	1	ID der Kommunikation
typ	String (Code)	1	Typ der Kommunikationsmethode Referenz auf Liste von Typen der Codeliste 12.19 Erreichbarkeitstyp
kennung	String (UTF-8)	1	Konkrete Angabe der zum Erreichen der Person oder Organisation notwendigen Information wie E-Mail-Adresse oder Telefonnummer.
revision	String (UTF-8)	1	Revisionsnummer der Kommunikation wird vom SchulConneX-Server mit der Erstellung des Datensatzes sowie Aktualisierung generiert Dieser Wert kann nicht von Quellsystemen oder Diensten gesetzt werden.

Datenmodell 9: Datenmodell Kommunikation

### Beispiel JSON eines Arrays mit mehreren Erreichbarkeiten des Datentyps Kommunikation

```
{  
    "id": "479a2b4a-1775-4481-809d-e22aa7384f13",  
    "typ": "EMail",  
    "kennung": "Petra.Muster@gymnasium-lehrte.de",  
}
```

Quellcode 6: Beispiel JSON des Datentyps Kommunikation

## 5.11 Beziehung

Das Datenmodell Beziehung beschreibt die Beziehung zweier Personen. Die Beziehungen werden zwischen Personenkontexten der beiden Personen erstellt. Beziehungen werden immer in der Form "hat als" dargestellt. Wenn aus einem Personenkontext von Person A eine Beziehung "SchB" (Schulbegleiter oder Schulbegleiterin) zu einem Personenkontext (ktid) von Person B besteht, so **hat** A die Person B **als** Schulbegleiter oder Schulbegleiterin und Person A **ist** schulbegleitet **von** Person B.

Attribut	Typ	Anz.	Bemerkung
id	String (UTF-8)	1	ID der Beziehung
ktid	String (UTF-8)	1	Id des Personenkontextes, zu dem eine Beziehung besteht
beziehung	String (UTF-8)	1	Beziehung aus Codeliste 12.20 Beziehungen
revision	String (UTF-8)	1	Revision der Beziehung wird vom SchulConneX-Server mit der Erstellung des Datensatzes sowie Aktualisierung generiert  Dieser Wert kann nicht von Quellsystemen oder Diensten gesetzt werden.

Datenmodell 10: Datenmodell Beziehung

### Beispiel JSON einer Beziehung

```
{
  "id": "4de182f3-4a8b-468b-9050-76dd1cae95f3",
  "ktid": "553f984f-5fea-48c8-ae26-089420465803",
  "beziehung": "SchB",
  "revision": "1.0"
}
```

Quellcode 7: Beispiel JSON für eine Beziehung



## 5.12 Anschrift

Die Anschrift erlaubt es, einen Ort zu beschreiben, an dem, beispielsweise, eine Organisation ansässig ist.

Attribut	Typ	Anz.	Bemerkung
postleitzahl	String (UTF-8)	0..1	Postleitzahl
ort	String (UTF-8)	0..1	amtlicher Gemeindename
ortsteil	String (UTF-8)	0..1	Teil eines Ortes, falls genauere Untergliederung gewünscht wird

Datenmodell 11: Datenmodell Anschrift

### Beispiel JSON einer Anschrift

```
{
  "postleitzahl": 80469,
  "ort": "München",
  "ortsteil": "Ludwigsvorstadt-Isarvorstadt",
}
```

Quellcode 8: Beispiel JSON für eine Anschrift

## 5.13 Organisationsbeziehung

Das Datenmodell Organisationsbeziehung beschreibt die Beziehung zweier Organisationen.

Attribut	Typ	Anz.	Bemerkung
orgid	String (UTF-8)	1	Id der Organisation, zu der die ‚hat als‘-Beziehung besteht
organisationsbeziehung	String (UTF-8)	1..n	Beziehung aus Codeliste 12.21 Organisationsbeziehungen

### Beispiel JSON einer Organisationsbeziehung

```
{  
  "orgid": "602a2034-bf6b-483c-bfa1-f3b02f67018f",  
  "beziehung": "SchBeh"  
}
```

*Quellcode 9: Beispiel JSON für eine Organisationsbeziehung*

## 6 Angepasstes Datenmodell für Dienstanbieter-Schnittstelle

Der Datenabruf durch einen Dienst erfolgt im Kontext eines Nutzers oder einer Nutzerin und stellt nur die Daten, die für die Nutzung des Dienstes erforderlich sind, bereit. Die Daten haben aus diesem Grund eine andere Struktur als an den Endpunkten für Quellsysteme. Außerdem sind nur lesende Abrufe (GET) möglich. Zur Referenzierung der Daten einer spezifischen Person oder von Personenkontexten wird eine pseudonymisierte ID (pid) genutzt, welche aus Sicht der einzelnen Dienste jeweils konstant und eindeutig ist. Diese pseudonymisierte ID entspricht nicht notwendig der intern in der Datenhaltung des SchulConneX-Services genutzten ID. Sie ist auch nicht identisch mit der pseudonymisierten ID für dieselbe Person oder denselben Personenkontext aus Sicht von anderen Diensten, um einen automatischen direkten Abgleich der Informationen zwischen zwei Anbietern, unter Umgehung des SchulConneX-Services, beispielsweise durch Korrelationsvermeidung - als Teil einer Strategie zur Minimierung der Möglichkeit, Benutzeridentitäten über Token-Empfänger hinweg zu verknüpfen - zumindest zu erschweren.

Nicht alle Attribute stehen allen Diensten zur Verfügung. Welche Daten vom SchulConneX-Server an den Dienst geliefert werden, wird durch individuelle Absprachen zwischen Dienstanbieter und Betreiber des SchulConneX-Servers festgelegt.

### **Hinweis zur Notation 0/1, 0/0..1 und 0/0..n bei der Anzahl**

Bei Attributen mit der Anzahl 0/1 hängt es von der vertraglichen Absprache zwischen Dienstanbieter und Betreiber des SchulConneX-Servers ab, ob dieses Attribut geliefert wird. Sieht die Absprache die Bereitstellung dieses Attributes nicht vor, so wird es **nie** geliefert (d. h. bei nicht erfolgter Freigabe ist die Anzahl als '0' zu lesen und das Vorhandensein des Attributes ist als Fehler zu interpretieren).

Sieht die Absprache die Bereitstellung dieses Attributes vor, so zeigt die Angabe nach dem Schrägstrich, ob ein Attribut immer (0/1) oder maximal einmal zu erwarten ist (0/0..1). Die Schreibweise 0/0..n zeigt an, dass kein Wert oder ein Array von Werten zu erwarten sind. Beispielsweise ist nicht für alle Personen ein Geburtsdatum hinterlegt, so dass, selbst wenn das Attribut "geburt.datum" freigegeben ist, nicht zwingend ein Geburtsdatum geliefert wird.

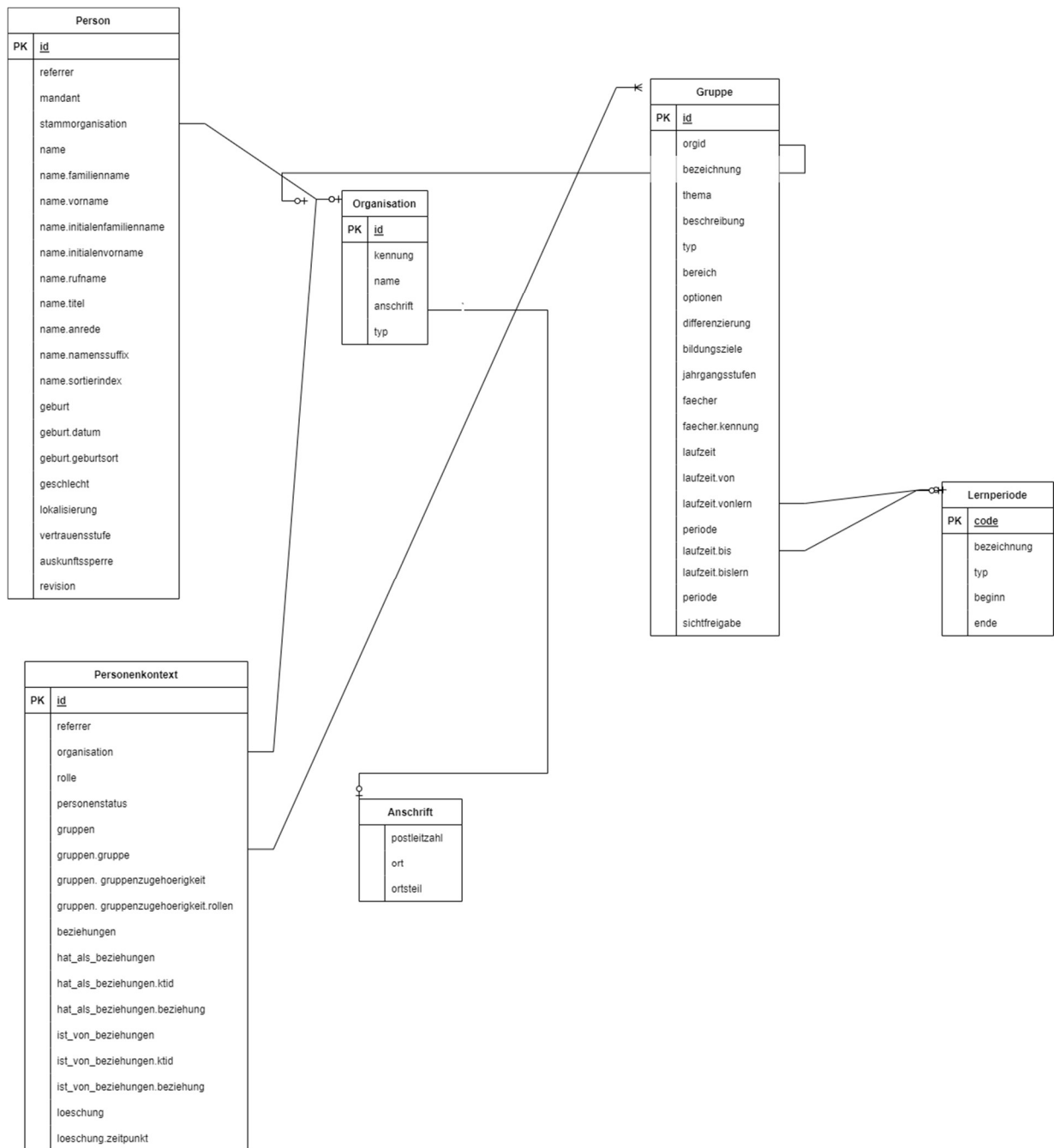


Abbildung 2: Graphische Darstellung des vereinfachten Datenmodells für Dienste

## 6.1 Angepasstes Datenmodell Person für Dienstanbieter

Attribut	Typ	Anz.	Freig. erford erlich	Bemerkung
<b>referrer</b>	String (UTF-8)	0..1	ja	ID der Person im Quellsystem wird vom Quellsystem vergeben und muss im Quellsystem eindeutig sein
<b>stammorganisation</b>	String (UTF-8)	0/0..1	ja	Personen können einer Organisation angehören, jedoch zeitweise an einer anderen Organisation tätig sein. Die erste Organisation ist die Stammorganisation. Sie ist unabhängig vom Personenkontext.
<b>stammorganisation.id</b>	String (UTF-8)	0/1	ja	ID der Stammorganisation wird vom SchulConneX-Server vergeben und ist eindeutig Dieses Attribut ist unveränderbar (immutable).
<b>stammorganisation.kennung</b>	String (UTF-8)	0/0..1	ja	Die optionale Kennung (Identifikations-ID) der Stammorganisation ist innerhalb eines Organisationstyps eindeutig
<b>stammorganisation.name</b>	String (DIN 91379.B)	0/0..1	ja	Offizieller Name der Stammorganisation
<b>stammorganisation.anschrift</b>	Anschrift	0/0..1	ja	Anschrift der Stammorganisation nach Datenmodell 5.12 Anschrift
<b>stammorganisation.typ</b>	String (Code)	0/0..1	ja	Typ der Stammorganisation Referenz auf einen Code der Codeliste 12.5 Organisationstyp
<b>name</b>	<i>Strukturierung</i>	0/1	ja	
<b>name.familienname</b>	String	0/1	ja	Familienname(n) der Person

Attribut	Typ	Anz.	Freig. erford erlich	Bemerkung
	(DIN 91379.A <sup>1</sup> )			
<b>name.vorname</b>	String (DIN 91379.A)	0/1	ja	Mehrere Vornamen werden durch Leerzeichen separiert.
<b>name.initialenvorname</b>	String (DIN 91379.A)	0/0..1	ja	Initial oder Initialen des Vornamens
<b>name.initialenfamiliennamenname</b>	String (DIN 91379.A)	0/0..1	ja	Initial oder Initialen des Familiennamens
<b>kommunikation</b>	Kommunikation[]	0..1	ja	Array von Kommunikations-Daten (siehe Datenmodell 5.10 Kommunikation) über welche die Person erreichbar ist
<b>geburt</b>	Strukturierung	0/0..1	ja	
<b>geburt.datum</b>	Datum (siehe Abschnitt 5.1)	0/0..1	ja	Geburtsdatum der Person
<b>geburt.volljaehrig</b>	String (Code)	0/0..1	ja	gibt an, ob eine Person volljährig ist, ohne weitere Information über Geburtsdatum oder Alter zu vermitteln  Referenz auf einen Code der Codeliste 12.7 Boolean
<b>geburt.geburtsort</b>	String (DIN 91379.A)	0/0..1	ja	Geburtsort einer Person, bestehend aus Ortsnamen und Land, beispielsweise "Eindhoven, Niederlande"

---

<sup>1</sup> DIN 91379.A referenziert die Spezifikation String.Latin+ 1.2 in der Ausprägung Datentyp A

Attribut	Typ	Anz.	Freig. erford erlich	Bemerkung
				Wird kein Land angegeben, so wird "Deutschland" angenommen.
<b>geschlecht</b>	String (Code)	0/0..1	ja	Referenz auf einen Code der Codeliste 12.2 Geschlecht
<b>lokalisierung</b>	String (Code)	0/0..1	ja	Referenz auf einen Code der Codeliste 12.9 Lokalisierung  bevorzugte Lokalisierungseinstellung einer Person für Anwendungen  Wenn nichts angegeben wurde, wird „de-DE“ zurückgeliefert.
<b>vertrauensstufe</b>	String (Code)	0/0..1	ja	gibt an, auf welchem Vertrauensniveau die Personendaten von der erfassenden Mandantschaft verifiziert wurden  Referenz auf einen Code der Codeliste 12.4 Vertrauensstufe

Datenmodell 13: Datenmodell Person für Dienstanbieter

## 6.2 Angepasstes Datenmodell Personenkontext für Dienstanbieter

Hat eine Person mehrere Personenkontexte, so wird typischerweise bereits bei der Anmeldung im Sicherheitskontext einer angemeldeten Person einer dieser Kontexte selektiert. In diesem Fall wird auch nur dieser ausgewählte Personenkontext an den Dienst weitergereicht.

Abhängig von Vereinbarungen mit dem Betrieb des SchulConneX-Servers ist es auch möglich, alle gespeicherten Personenkontexte mitzuliefern und die Auswahl erst in der Anwendung des Dienstes vorzunehmen. Dies ist beispielsweise sinnvoll, wenn Informationen aus einem Kontext in einen anderen Kontext übernommen werden sollen, beispielsweise bei Schulwechsel.

Attribut	Typ	Anz.	Attr .- Frei g. erfo	Bemerkung

			rder lich	
<b>referrer</b>	String (UTF-8)	0..1	ja	ID der Gruppe im Quellsystem wird vom Quellsystem vergeben und muss im Quellsystem eindeutig sein
<b>id</b>	String	1	nein	ID des Personenkontexts
<b>organisation</b>	<i>Strukturierung</i>	0/1	ja	Beschreibung der Organisation siehe 6.3 Angepasstes Datenmodell Organisation für Dienstanbieter
<b>rolle</b>	String (Code)	0/1	ja	Rolle der Person innerhalb der Organisation  Referenz auf einen Code der Codeliste 12.3  Rolle
<b>personenstatus</b>	String (Code)	0/1	ja	Status, den eine Person in einer Organisation in Bezug auf eine bestimmte Rolle hat  Referenz auf einen Code der Codeliste 12.1 Personenstatus
<b>gruppen</b>	<i>Strukturierung</i>	0/0..n	ja	<i>Array aus Gruppen und dazugehörige Gruppenzugehörigkeiten</i>
<b>gruppen.gruppe.*</b>	<i>Komplexes Objekt</i>	0/1	ja	<i>Attribute einer einzelnen Gruppe entsprechend Datenmodell Gruppe 5.6.</i>
<b>gruppen. gruppenzugehoerigkeit</b>	<i>Strukturierung</i>	0/1	ja	
<b>gruppen. gruppenzugehoerigkeit. rollen</b>	String (Code)	0/0..n	ja	Rollen der Person innerhalb der Gruppe  Liste von Rollen nach Codeliste 12.14 Gruppenrolle
<b>gruppen. sonstige_gruppenzuge hoerige</b>	Array {String (Code), String (Code)}	0/0..1	ja	weitere Zugehoerige zu der Gruppe, jeweils als Paare von Kontext-IDs ktid und Rollen (Array von Rollen nach Codeliste 12.14 Gruppenrolle)



<b>beziehungen</b>	<i>Strukturierung</i>	0/0..1	ja	<i>Array aus 'hat' und ,ist Beziehungen</i>
<b>hat_als_beziehungen</b>	<i>Strukturierung</i>	0/0..1	ja	<i>Array der 'hat als' Beziehungen eines Personenkontextes</i> <i>Jeder Eintrag enthält eine Personenkontext-Id und eine Beziehung.</i>
<b>hat_als_beziehungen.ktid</b>	String (UTF-8)	0/1	ja	pseudonymisierte ID des Personenkontexts zu dem aus dem aktuellen Personenkontext eine ,hat' Beziehung besteht
<b>hat_als_beziehungen.beziehung</b>	String (Code)	0/1	ja	Art der Beziehung aus Codeliste 12.20 Beziehungen
<b>ist_von_beziehungen</b>	<i>Strukturierung</i>	0/0..1	ja	<i>Array der 'ist von' Beziehungen eines Personenkontextes</i> <i>Jeder Eintrag enthält eine Personenkontext-Id und eine Beziehung.</i>
<b>ist_von_beziehungen.ktid</b>	String (UTF-8)	0/1	ja	pseudonymisierte ID des Personenkontexts zu dem aus dem aktuellen Personenkontext eine ,ist' Beziehung besteht
<b>ist_von_beziehungen.beziehung</b>	String (Code)	0/1	ja	Art der Beziehung aus Codeliste 12.20 Beziehungen
<b>loeschung</b>	<i>Strukturierung</i>	0/0..1	nein	
<b>loeschung.zeitpunkt</b>	String (datetime)	1	nein	Datum und Uhrzeit der Löschung des Personenkontexts  Das Format des Löschezitpunktes ist yyyy-MM-dd'T'hh:mm'Z' als UTC-Zeitpunkt.

Datenmodell 14: Datenmodell Personenkontext für Dienstleister

### 6.3 Angepasstes Datenmodell Organisation für Dienstanbieter

Attribut	Typ	Anz.	Attr.-Freig. erforderlich <sup>2</sup>	Bemerkung
<b>id</b>	String (UTF-8)	1	nein	ID der Organisation  wird vom SchulConneX-Server vergeben und ist eindeutig. Dieses Attribut ist unveränderbar (immutable).
<b>kennung</b>	String (UTF-8)	0/0..1	ja	Die optionale Kennung (Identifikations-ID) einer "Organisation" muss innerhalb eines Organisationstyps eindeutig sein.
<b>name</b>	String (DIN 91379.B)	0/0..1	ja	offizieller Name einer Organisation
<b>anschrift</b>	Anschrift	0/0..1	ja	Anschrift der Organisation
<b>typ</b>	String (Code)	0/0..1	ja	Typ der Organisation. Referenz auf einen Code der Codeliste 12.5 Organisationstyp

Datenmodell 15: Datenmodell Organisation für Dienstanbieter

---

<sup>2</sup> Die Konfigurierbarkeit durch Absprache mit dem SchulConneX Server-Betreiber dient an dieser Stelle nicht dem Schutz personenbezogener Daten, sondern zur Reduktion von Bandbreite, wenn Informationen von Diensten nicht benötigt werden.

## 6.4 Angepasstes Datenmodell Gruppe für Dienstanbieter

Eine Gruppe erlaubt es, mehrere Personen in einer Gruppe zusammenzufassen. Typischerweise handelt es sich bei einer Gruppe um eine Schulklasse oder eine ähnliche Struktur in einer Schule oder anderen Organisation. Die Zugehörigkeit einer Person zu einer Gruppe wird durch das Datenmodell Gruppenzugehörigkeit beziehungsweise zusammen mit der Gruppen-Information im Gruppendatensatz abgebildet.

Attribut	Typ	Anz.	Attr.-Freig. erfor- derlich	Bemerkung
id	String (UTF-8)	1	nein	ID der Gruppe  wird vom SchulConneX-Server vergeben und ist eindeutig
orgid	String (UTF-8)	0/0..1	ja	ID der Organisation, welcher die Gruppe zugeordnet ist
bezeichnung	String (UTF-8)	0/1	ja	beschreibt die Bezeichnung der Gruppe, beispielsweise „Englisch, 2. Klasse“
thema	String (UTF-8)	0/0..1	ja	Thema der Gruppe in Kurzform, idealerweise <35 Zeichen
beschreibung	String (1024) (UTF-8)	0/0..1	ja	zusätzliche optionale Beschreibung der Gruppe
typ	String (Code)	0/1	ja	Typ der Gruppe (Klasse, Kurs)  Referenz auf Code der Codeliste 12.13 Gruppentyp
bereich	String (Code)	0/0..1	ja	Art der Gruppe (Pflicht, Wahl, Grundkurs, Leistungskurs...)  Referenz auf Code der Codeliste 12.10 Gruppenbereich
optionen	String (Code)	0/0..n	ja	zusätzliche Optionen zur Gruppe, meistens

				sprachbezogen (bilingual, herkunftssprachlich)  Referenz auf Code der Codeliste 12.12 Gruppenoption
differenzierung	String (Code)	0/0..1	ja	Differenzierung bei integriertem Unterricht oder in der gymnasialen Oberstufe (Erweiterungsebene, Grundebene)  Referenz auf Code der Codeliste 12.11 Gruppendifferenzierung
bildungsziele	String (Code)	0/0..n	ja	Bildungsziel (Realschule, Hauptschule, Gymnasium, ...). Referenz auf Code der Codeliste 12.18 Bildungsziel
jahrgangsstufen	String (Code)	0/0..n	ja	Jahrgangsstufe (Jahrgangsstufe 1 bis Jahrgangsstufe 10)  Referenz auf Code der Codeliste 12.6 Jahrgangsstufe
faecher	Strukturierung	0/0..n	ja	Liste der in der Gruppe behandelten Fächer
faecher.kennung	String (Code)	0/0..1	ja	in der Gruppe behandelte Fächer  Referenz auf Code der Codeliste 12.17 Fächerkanon
laufzeit	<i>Strukturierung</i>	0/1	ja	Die Laufzeiten von Gruppen können entweder direkt durch Datumsangaben festgelegt werden (von/bis) oder durch Referenzen auf Lernperioden (vonlernperiode/bislernperiode).

laufzeit.von	Datum (siehe Abschnitt 5.1)	0/0..1	ja	Anfang des Gültigkeitszeitraumes der Gruppe  Dieser Zeitpunkt kann auch in der Zukunft liegen.
laufzeit.vonlernperiode	String (Code)	0/0..1	ja	Code der zu einer Gruppe gehörenden Lernperiode  Referenz auf Code der Codeliste 12.15 Lernperiode
laufzeit.bis	Datum (siehe Abschnitt 5.1)	0/0..1	ja	Ende des Gültigkeitszeitraumes der Gruppe
laufzeit.bislernperiode	String (Code)	0/0..1	ja	Code der zu einer Gruppe gehörenden Lernperiode  Referenz auf Code der Codeliste 12.15 Lernperiode
sichtfreigabe	String (Code)	0/0..1	ja	gibt an, ob diese Gruppe aufgrund der Freigabe durch eine andere Organisation sichtbar ist  Ist "sichtfreigabe" nicht gesetzt, so entspricht das dem Wert "nein". Der Wert von Sichtfreigabe ist Boolean nach Codetabelle 12.7 Boolean.

*Datenmodell 16: Angepasstes Datenmodell Gruppe für Dienstanbieter*

## 7 Schnittstellendefinition

### 7.1 Allgemeine Festlegungen

#### 7.1.1 Standard-Anfragen (Default Requests)

Der Webserver bietet Webclients die Möglichkeit, durch standardisierte HTTP-Methoden gespeicherte Ressourcen bzw. Entitäten abzurufen bzw. zu bearbeiten.

##### *GET*

GET ruft eine oder mehrere Entitäten/Ressourcen vollständig oder teilweise ab. Der GET-Anfrage können zusätzliche Informationen mitgegeben werden, die der Webserver verarbeiten soll. Diese URL-Parameter werden an die URL angehängt.

##### *POST*

Per POST können große Datenmengen zum Webserver gesendet werden. Dabei werden die URL-Parameter nicht in die URL, sondern in den HTTP-Header geschrieben. POST wird oft bei Formularen benutzt, um mehrere Daten zu senden bzw. Entitäten zu erstellen.

##### *PUT*

PUT funktioniert ähnlich wie POST und dient dazu, Dateien auf dem Webserver abzulegen bzw. bereits existierende Entitäten zu verändern.

#### 7.1.2 Standard-Rückgaben (Default Responses)

Nachdem der Webserver eine Anfrage erhalten und interpretiert hat, wird mit einer HTTP-Response geantwortet. Eine HTTP-Response besitzt einen HTTP-Status-Code. Die erste Ziffer des Status-Codes gibt die Kategorie der HTTP-Response an:

1xx: Informative Antworten

2xx: Erfolg

3xx: Weiterleitung

4xx: Client-Fehler

5xx: Server-Fehler

## 7.2 Erfolgreiche Ausführung

### *200 (Erfolgreiche Ausführung)*

Die angeforderten Daten werden mit der HTTP-Response übermittelt, beispielsweise werden die Daten einer Person im JSON-Datenformat übergeben.

### *201 (Erstellt)*

Eine neue Resource wurde erstellt.

## 7.3 Daten unverändert

### *304 (Not Modified)*

Die angeforderten Daten wurden seit dem letzten Zugriff nicht verändert und nicht erneut ausgeliefert. Dieser Rückgabewert erfolgt nur dann, wenn beim vorhergehenden Zugriff auf den Datensatz ein ETag geliefert wurde und beim erneuten Zugriff If-None-Match mit dem ETag im HTTP-Anfrage-Header benutzt wurde sowie der aktuelle ETag des Servers und der ETag im HTTP-Anfrage-Header übereinstimmen.

## 7.4 Fehlerbehandlung

Fehler bei der Nutzung der Schnittstelle werden grundsätzlich mit einem der unten gelisteten Response Codes zurückgegeben. Zusätzlich besitzt die Response auch einen Error Payload, welcher mehr Details über den Fehler enthält und für Entwickler gedacht ist.

Format des Error Payloads mit Beispielwerten:

```
{
  "code": "401",
  "subcode": "01",
  "titel": " Access Token abgelaufen",
  "beschreibung": "Der Access-Token ist abgelaufen und muss erneuert werden."
}
```

*Quellcode 10: Beispiel des Error Payloads*

### *400 (Bad request)*

Der Aufruf ist nicht vollständig bzw. fehlerhaft, so dass die Bearbeitung nicht durchgeführt werden kann. Die Angaben beim Aufruf müssen vervollständigt bzw. angepasst werden.

Code	Sub-code	Titel	Beschreibung
400	00	„Fehlerhafte Anfrage“	„Die Anfrage ist fehlerhaft:“ + spezifischer Hinweis auf erkannten Fehler
400	01	„Fehlende Parameter“	„Folgende Parameter fehlen:“ + Liste fehlender Parameter
400	02	„Falsche Parameter“	„Folgende Parameter haben andere Werte als von der Schnittstelle erwartet:“ + Liste fehlerhafte Parameter
400	03	„Validierungsfehler“	„Die Anfrage konnte nicht erfolgreich validiert werden.“ + spezifische Fehlerbeschreibung, weshalb die Validierung fehlgeschlagen ist  Dies ist ein genereller Validierungsfehler, sofern keine der folgenden Fehlermeldungen passen.
400	04	„JSON-Struktur ungültig“	Der Payload entspricht keiner gültigen JSON-Struktur.
400	05	„JSON-Struktur nicht deserialisierbar“	Payload ist nicht deserialisierbar.
400	06	„JSON-Struktur besitzt ungültige Attribute“	unbekannte, beziehungsweise nicht gültige, Attribute vorhanden
400	07	„Attributwerte haben eine ungültige Länge“	„Textlänge von Attribut x ist nicht valide“, wobei x der Attributname ist, bspw. person.name
400	08	„Attributwerte entsprechen nicht dem gültigen Zeichensatz“	„Text von Attribut X entspricht nicht dem Zeichensatz Y“, wobei X der Attributname ist, bspw. person.name und Y der Zeichensatz, bspw. UTF-8



Code	Sub-code	Titel	Beschreibung
400	09	„Datumsattribut hat einen ungültigen Wert“	„Datumsformat von Attribut X ist ungültig“, wobei X der Attributname ist
400	10	„Attributwerte entspricht keinem der erwarteten Werte“	„Attribut X muss einen gültigen Wert aus der Werteliste für Attribut X enthalten.“, wenn Attribut X keinen Wert oder einen ungültigen Wert zugewiesen bekommen hat, wobei X der Attributname ist
400	11	„Attribut darf nicht mit diesem Wert gesetzt oder verändert werden.“	„Attribut x darf aufgrund fehlender Berechtigung nicht mit diesem Wert gesetzt oder verändert werden“, wobei x der Attributname ist, bspw. person.rolle
400	12	„Person enthält noch Personenkontexte.“	„Daten vom Typ "Person" können nur gelöscht werden, wenn für diese Person keine Personenkontexte mehr existieren.“
400	13	„Personenkontext wird genutzt.“	„Personenkontexte können über die API /personenkontexte/{id} nur dann direkt gelöscht werden, wenn sie von keinem anderen System genutzt wurden. Wurde der Personenkontext bereits extern genutzt, so muss die Löschung über andere APIs erfolgen. „
400	14	„Zyklische Referenzgruppe“	„Referenzgruppen dürfen keine zirkulären Referenzen haben. „
400	15	„Text zu lang“	„Die Länge eines übergebenen Textattributes überschreitet die in der Spezifikation angegebene Maximallänge.“
400	16	„Inkonsistente Laufzeitangabe“	„Laufzeiten (von Lernperioden) müssen entweder als von/bis oder als vonlernperiode/bislernperiode

Code	Sub-code	Titel	Beschreibung
			angegeben werden. Eine Mischung (von/bislernperiode oder vonlernperiode/bis) ist nicht zulässig.“
400	17	"Doppelter Filter"	„Jeder Filter darf in der URL nur einmal benutzt werden. Filter wie ?pid=123&pid=124 sind nicht zulässig.“
400	18	"Beziehung darf so nicht erstellt werden."	„Die Beziehung zwischen zwei Personenkontexten darf so nicht erzeugt werden.“ + spezifische Fehlermeldung der Schnittstelle
400	19	"Erreichbarkeit kann so nicht hinzugefügt werden.“	„Diese Erreichbarkeit ist so nicht zulässig.“ + spezifische Fehlermeldung der Schnittstelle

Fehlerliste 1: Fehler 400 Bad Request

#### 401 (Unauthorized)

Eine Autorisierung wurde nicht angegeben bzw. kann nicht bestätigt werden. Es ist auch möglich, dass die Authentifizierung erneut durchgeführt werden muss, wenn die Sitzung abgelaufen ist.

Code	Sub-code	Titel	Beschreibung
401	00	„Zugang verweigert“	„Die Anfrage konnte aufgrund fehlender Autorisierung nicht verarbeitet werden.“ + spezifische Fehlermeldung der Schnittstelle, die nicht in eine der anderen Kategorien fällt
401	01	„Access Token abgelaufen“	„Der Access-Token ist abgelaufen und muss erneuert werden.“
401	02	„Invalid Access-Token“	„Invalid Access-Token. Autorisierung fehlgeschlagen“ Dieser Fehler tritt auf,

Code	Sub-code	Titel	Beschreibung
			wenn der Access-Token nicht valide ist oder nicht vollständig.
401	03	„Falsche Autorisierungsmethode“	„Die Anfrage konnte aufgrund einer nicht unterstützten Autorisierungsmethode nicht verarbeitet werden“ + spezifische Fehlermeldung der Schnittstelle, welche Autorisierungsmethode benutzt werden kann, bspw. „Authorization: Bearer“

*Fehlerliste 2: Fehler 401 Unauthorized*

### 403 (Forbidden)

Der Aufruf wird nicht ausgeführt, da nicht ausreichend Rechte vorhanden sind.

Die mit dem Token von der Authentifizierung übergebenen Zugriffsrechte sind nicht ausreichend, um den Aufruf vollständig auszuführen.

Code	Sub-code	Titel	Beschreibung
403	00	„Fehlende Rechte“	„Die Autorisierung war erfolgreich, aber die erforderlichen Rechte für die Nutzung dieses Endpunktes sind nicht vorhanden.“ + spezifische Fehlermeldung der Schnittstelle, die nicht in eine der anderen Kategorien fällt

*Fehlerliste 3: Fehler 403 Forbidden*

### 404 (Not found)

Der Endpunkt existiert nicht bzw. eine geforderte Entität ist nicht verfügbar.

Code	Sub-code	Titel	Beschreibung
404	00	„Endpunkt existiert nicht“	„Der aufgerufene Endpunkt existiert nicht.“

404	01	„Angefragte Entität existiert nicht“	„Die angeforderte Entität existiert nicht.“
-----	----	--------------------------------------	---

*Fehlerliste 4: Fehler 404 Not found*

#### 405 (Method not allowed)

Die konkrete Anfrage ist nicht erlaubt.

Code	Sub-code	Titel	Beschreibung
405	00	„Nicht erlaubt“	„Dieser Aufruf ist nicht erlaubt“ + spezifische Fehlermeldung der Schnittstelle, die nicht in eine andere Kategorie fällt
405	01	„POST/PUT nicht erlaubt“	„Für diesen Endpunkt ist ein POST/PUT nicht erlaubt.“

*Fehlerliste 5: Fehler 405 Method not allowed*

#### 409 (Conflict)

Die zu verändernde Entität wurde durch Dritte verändert und führt daher zu einem Versionskonflikt.

Code	Sub-code	Titel	Beschreibung
409	00	„Konflikt mit dem aktuellen Zustand der Resource.“	„Die Entität wurde eventuell durch Dritte verändert. Die Revisionsnummer stimmt nicht überein.“

*Fehlerliste 6: Fehler 409 Conflict*

#### 500 (Internal Serverfehler)

Aufgrund eines Serverfehlers kann der Aufruf nicht ausgeführt werden.

Code	Sub-code	Titel	Beschreibung
500	00	„Interner Serverfehler“	„Es ist ein interner Fehler aufgetreten.“ + spezifische Fehlermeldung der Schnittstelle,

			die nicht in eine der anderen Kategorien fällt
--	--	--	--

*Fehlerliste 7: Fehler 500 Interner Serverfehler*

## 7.5 Validierung der Anfrage

Alle Anfragen werden validiert und auf Korrektheit überprüft. Sobald eine Anfrage die Validierung nicht besteht, wird eine Fehlermeldung zurückgegeben.

Die Priorisierung der Validierung sieht wie folgt aus:

Zunächst wird die Authentizität überprüft und ein „401 Unauthorized“ zurückgegeben, sobald eine Autorisierung benötigt wird.

Anschließend wird das Datenmodell in der Anfrage überprüft. Bei dieser Validierung wird beim ersten Fehler im Datenmodell sofort mit einem passenden Fehler geantwortet und die Verarbeitung der Anfrage abgebrochen. Fehlermeldungen sind unter Kapitel „7.4 Fehlerbehandlung“ – „400 Bad Request“ – Subcode 03 – „Validierungsfehler“ gelistet.

Bei PUT und CREATE Anfragen werden Query Parameter dekodiert und wie Attributwerte des Payloads validiert. Query Parameter werden nur URL-kodiert angenommen.

## 8 Schnittstellen für Quellsysteme und Dienste

Die von SchulConneX Servern bereitgestellten Endpunkte sind als REST-API realisiert. Die spezifischen Funktionen der einzelnen Endpunkte werden in den Kapiteln 9 API-Endpunkte für Dienstanbieter und 10 API-Endpunkte für Quellsysteme beschrieben.

Allgemein erfolgen Zugriffe auf Ressourcen über die Endpunkte mit Hilfen von CRUD-Operationen (CREATE, READ, UPDATE und DELETE). Welche Zugriffe jeweils erlaubt sind und welche HTTP(S) Methode dabei genutzt werden dürfen ist bei den Endpunktbeschreibungen im Detail aufgeführt. Dort werden auch die zu verwendenden JSON Strukturen beschrieben.

### 8.1 API-Versionierung

Die URL der Schnittstelle besteht aus einer Base-URL (nachfolgend fett markiert) sowie dem eigentlichen Endpunkt.

`https://subdomain.domain.org/subpaths/endpoint`

*Quellcode 11: URL-Struktur der Schnittstelle*

Die API-Versionierung ermöglicht es den Entwicklern des SchulConneX Servers, Anpassungen an der API zu implementieren und gleichzeitig zu gewährleisten, dass Clients mit aktiver API-Anbindung weiterhin fehlerfrei funktionieren. Zusätzlich wird gewährleistet, dass Clients genug Zeit zur Implementierung einer neuen Version der API erhalten.

Die Versionierung der API befindet sich in der URL zur REST-API. Dabei wird die Versionsnummer der API in der URL als Pfad am Ende der Base-URL mit angegeben. Beispiel: `base-url/v1/...`, wobei hier „v1“ für die API-Versionsnummer 1.0 steht. Eine mögliche API-Version 1.1 oder 2.0 wird wie folgt aufgerufen: `base-url/v1-1/...` bzw. `base-url/v2/...`

Im Folgenden ist ein Beispiel einer Schnittstellen-URL dargestellt.

`https://subdomain.domain.org/subpath /v1/organisation-info`

*Quellcode 12: Beispiel der URL eines Schnittstellenendpunkts*

Die konkrete URL für die Schnittstelle erhalten Sie von dem Betreiber des SchulConneX-Servers.

### 8.2 Registrierung

Gemäß der Spezifikation OAuth muss ein Dienst oder ein Quellsystem (client-Anwendung) gegenüber SchulConneX bekannt sein – diesbezüglich wird die client-Anwendung über einen organisatorischen Prozess registriert.

Durch die Registrierung wird dem Quellsystem eine Client-ID und das dazugehörige Client Secret zur Verfügung gestellt.

### 8.3 API-Autorisierung für Quellsysteme

Bei Quellsystemen (Client-Anwendung) authentifiziert und autorisiert der SchulConneX-Server die Client-Anwendung und nicht einen Benutzer. Hierfür verwenden Client-Anwendungen den OAuth 2.0 Client Credentials Flow (OAuth 2.0 RFC 6749<sup>3</sup> "The OAuth 2.0 Authorization Framework", Abschnitt 4.4), bei dem eine Client-Anwendung ihre Client-ID und das dazugehörige Client Secret verwendet, um sich zu authentifizieren und einen Access Token zu erhalten. Bei dem Access Token handelt es sich um eine Zeichenkette, welche von den Client-Anwendungen bei den darauffolgenden HTTP-Anfragen gegen die REST-API im Authorization-Header verwendet wird.

### 8.4 Autorisierung und Authentifikation bei Diensteanbietern

Die Autorisierung entspricht dem Standard OAuth-Modell und ist im folgenden Sequenz-Diagramm dargestellt. Danach werden die beiden Attribut-Listen „Access Token“ basierend auf der Spezifikation OAuth 2.0 Authorization Framework sowie „ID Token“ gemäß der Spezifikation OpenID Connect 1.0 definiert, jeweils mit spezifischen Anpassungen.

Für die Nutzung der in diesem Dokument spezifizierten Schnittstellenendpunkte (REST-APIs) des SchulConneX-Servers ist ein Access Token inklusive eines ID-Tokens notwendig. Dieser wird von einem separaten Authentication-Dienst ausgestellt. Den konkreten Endpunkt des Authentication-Dienstes erhalten Sie vom Betreiber des SchulConneX-Servers.

Für die Autorisierung eines Dienstes und die Authentifikation einer nutzenden Person gegenüber der REST-API wird das OAuth 2.0 Protokoll verwendet. Gemäß der Spezifikation IETF RFC 6749<sup>4</sup> „The OAuth 2.0 Authorization Framework“ muss der REST-API-Endpunkt für den „OAuth 2.0 Token Exchange“ (siehe IETF RFC 8693<sup>5</sup> "OAuth 2.0 Token Exchange") genutzt werden. Über die Schnittstelle wird einer Client-Anwendung im Erfolgsfall ein JSON mit Token ausgestellt; dieser beinhaltet unter anderem die Autorisierung eines Dienstes (den OAuth `access_token`).

---

<sup>3</sup> <https://datatracker.ietf.org/doc/html/rfc6749#section-4.4>

<sup>4</sup> <https://datatracker.ietf.org/doc/html/rfc6749>

<sup>5</sup> <https://datatracker.ietf.org/doc/html/rfc8693>

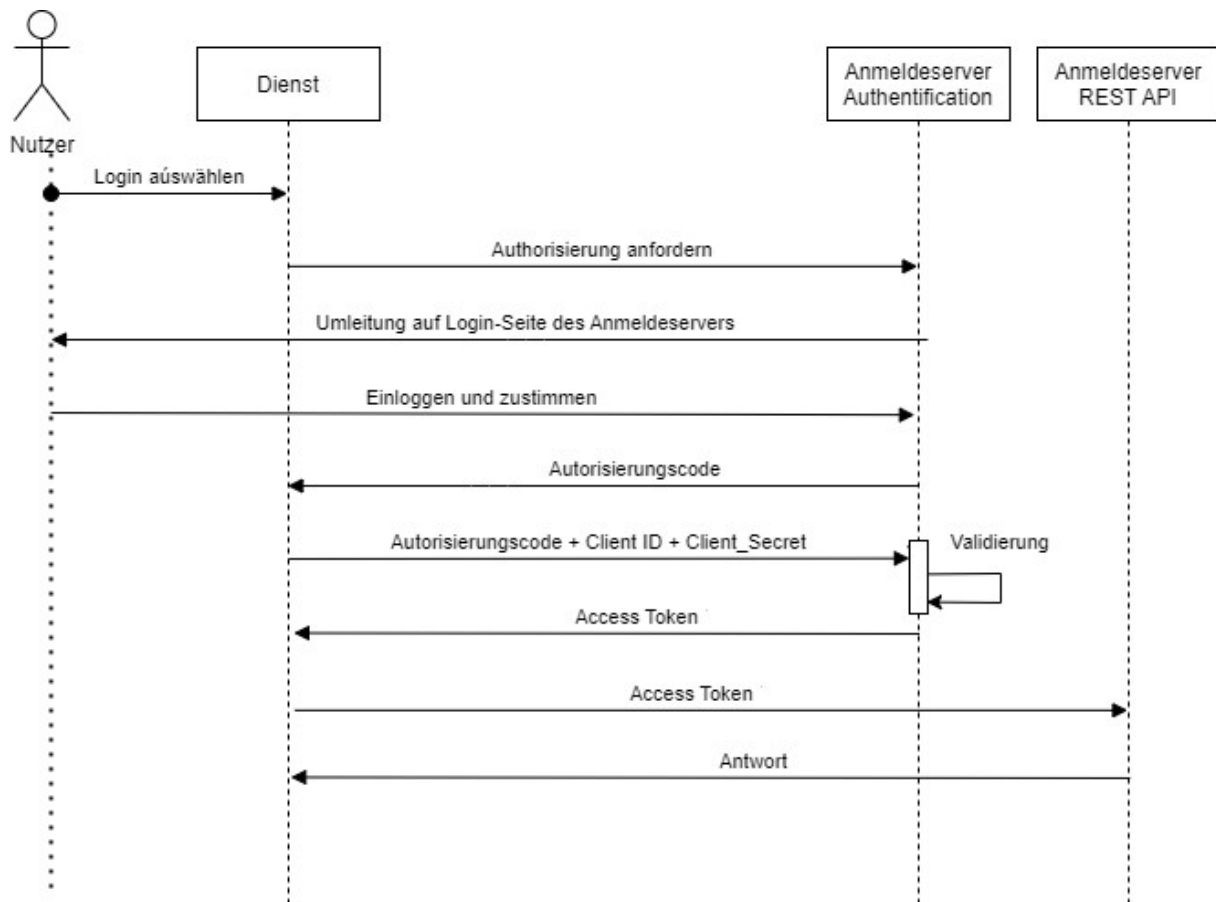


Abbildung 3: Sequenzdiagramm für die Autorisierung und anschließende Beispielabfrage von Personendaten im Sicherheitskontext einer angemeldeten nutzenden Person

Die Attribute für den erfolgreichen OAuth 2.0 Token Exchange ist im nachfolgenden Unterkapitel „Autorisierung und Authentifikation“ definiert. Bei dem `access_token` handelt es sich um eine Zugriffsberechtigung in Form einer Zeichenkette, welche von den Client-Anwendungen bei den darauffolgenden HTTP-Anfragen gegen die REST-API im Authorization Header verwendet wird (siehe Attribut `token_type`); dies ist in IETF RFC 6750 „The OAuth 2.0 Authorization Framework: Bearer Token Usage“<sup>6</sup> spezifiziert.

Das ID Token führt unter anderem das Attribut `sub` (kurz für subject identifier). Der Wert des Attributs `sub` entspricht einer für die Client-Anwendung pseudonymisierten ID `pid` der angemeldeten nutzenden Person im gewählten Sicherheitskontext. Das Datenmodell für den `id_token` ist in Kapitel 8.4.2 „ID Token“ beschrieben.

Dieser ID Token wird bei der Autorisierung eines Dienstes im Sicherheitskontext einer angemeldeten Nutzerperson (OAuth 2.0 Authorization Code Grant<sup>7</sup>) ausgestellt und als Attribut `id_token` im JSON der erfolgreichen OAuth 2.0 Token Exchange Anfrage geführt. Der für die Client-Anwendung pseudonymisierte Identifier (Id) der nutzenden Person

<sup>6</sup> <https://datatracker.ietf.org/doc/html/rfc6750>

<sup>7</sup> <https://datatracker.ietf.org/doc/html/rfc6749#section-1.3.1>



beschreibt die Person im Sicherheitskontext der Anmeldung und ist als Attribut `sub` im `id_token` enthalten.

In einigen Fällen ist es für Dienstanbieter wünschenswert, neben der PID bereits bei der Autorisierung weitere Informationen über die angemeldete Person zu erhalten. Dieses kann, nach Absprache mit dem Betreiber des SchulConneX-Servers, durch die Nutzung von OIDC Claims geschehen. Die Nutzung von OIDC Claims zur Übermittlung von SchulConneX-Informationen ist in Kapitel 9.2 "Person-Info über OIDC Claims" beschrieben. Im Regelfall sollten aber die REST APIs zur Abfrage von Informationen genutzt werden.

Für darauffolgende Anfragen der Client-Anwendungen gegen einen API-Endpunkt des SchulConneX-Services muss der `access_token` als Authorization Header im XMLHttpRequest (XHR) verwendet werden. Das Attribut `token_type` beim OAuth 2.0 Token Exchange Request referenziert die Art der Verwendung des `access_token` – gemäß IETF RFC 6750 „The OAuth 2.0 Authorization Framework: Bearer Token Usage“ oder den Basic<sup>8</sup> and Digest Authentifizierungsschemata.

Im Kontext der Spezifikation OpenID Connect *muss* das Attribut `token_type` den Wert „Bearer“ führen.

#### 8.4.1 Access Token

Die folgende Tabelle beschreibt die Attribute für die JSON-Antwort einer OAuth 2.0-Token-Exchange-Anfrage. Diese JSON-Struktur beinhaltet unter anderem die Autorisierung eines Dienstangebots (den OAuth `access_token`) und die Authentifikation einer nutzenden Person (den OpenID Connect `id_token`) . Beim Berechtigungstyp „OAuth 2.0 Client Credentials Grant“<sup>9</sup> wird vom Dienst ein Zugriffstoken (`access_token`) außerhalb des Sicherheitskontexts einer angemeldeten nutzenden Person angefordert. In diesem Fall wird kein `id_token` ausgestellt. Basierend auf der Spezifikation OAuth 2.0 ist der Antwort-Typ eines OAuth 2.0 Token Exchange ein JSON-Dokument (HTTP ContentType „application/json“), welches der IETF RFC8693-Spezifikation<sup>10</sup> entspricht.

Attribut Ebene 0	Typ	Anz.	Bemerkung
<code>access_token</code>	String (byte)	1	ein Token ( <code>access_token</code> ), welches vom Service auf die OAuth 2.0-Token-Exchange-Anfrage im Erfolgsfall ausgestellt wurde  Dieses ausgestellte <code>access_token</code> des zugrunde liegenden Authorization

---

<sup>8</sup> <https://www.rfc-editor.org/rfc/rfc7617>

<sup>9</sup> <https://datatracker.ietf.org/doc/html/rfc6749#section-4.4>

<sup>10</sup> <https://datatracker.ietf.org/doc/html/rfc8693>

Attribut Ebene 0	Typ	Anz.	Bemerkung
			Frameworks ist spezifiziert in IETF RFC6749 Abschnitt 5.1 <sup>11</sup> .
<b>expires_in</b>	Integer	0..1	<p>die Gültigkeit des Tokens (access_token) in Sekunden ab Ausstellung</p> <p>Das access_token beinhaltet ebenfalls die Dauer der Gültigkeit des Tokens. Dieser Wert ist <i>optional</i>, sofern ein access_token ausgestellt wurde.</p> <p>Beispielhaft steht der Wert 1800 für eine Gültigkeitsdauer von 30 Minuten.</p>
<b>refresh_token</b>	String (byte)	0..1	<p>ein Token (refresh_token), welches vom SchulConneX-Server auf die OAuth 2.0-Token-Exchange-Anfrage im Erfolgsfall <i>optional</i> ausgestellt wurde</p> <p>Mit dem refresh_token kann beim SchulConneX-Server ein neues access_token angefordert werden.</p>
<b>token_type</b>	String (Enumerated)	0..1	<p>Der token_type stellt die Information zur Verfügung, wie das access_token angewendet werden kann.</p> <p>Auswahl an token_type:</p> <ul style="list-style-type: none"> <li>• bearer<sup>12</sup></li> <li>• mac<sup>13</sup></li> </ul> <p>Im Kontext der Spezifikation OpenID Connect <i>muss</i> das Attribut token_type den Wert „Bearer“ führen.</p>
<b>id_token</b>	String (byte)	0..1	<p>Basierend auf der Spezifikation OpenID Connect beinhaltet die Token-Antwort einer OAuth 2.0-Token-Exchange-Anfrage zusätzlich ein Token (id_token<sup>14</sup>), welches Informationen über die Authentifizierung</p>

---

<sup>11</sup> <https://datatracker.ietf.org/doc/html/rfc6749#section-5.1>

<sup>12</sup> <https://datatracker.ietf.org/doc/html/rfc6750>

<sup>13</sup> <https://datatracker.ietf.org/doc/html/rfc6749#ref-OAuth-HTTP-MAC>

<sup>14</sup> [https://openid.net/specs/openid-connect-core-1\\_0.html#CodeIDToken](https://openid.net/specs/openid-connect-core-1_0.html#CodeIDToken)

Attribut Ebene 0	Typ	Anz.	Bemerkung
			einer nutzenden Person beinhaltet. Siehe 8.4.2
<b>scope</b>	String (byte)	0..1	eine Liste von durch Leerzeichen getrennten Zeichenketten, die den Geltungsbereich <sup>15</sup> des Tokens spezifiziert

Datenmodell 17: Access Token

#### 8.4.2 ID Token

Die folgende Tabelle beschreibt die Attribute für eine authentifizierte nutzende Person in Form eines sogenannten OpenID Connect ID-Tokens. Bei dem `id_token` handelt es sich um ein JSON Web Token (JWT)<sup>16</sup>. Von Interesse für Dienstanbieter ist das Attribut `sub` (subject identifier). Bei dieser Zeichenkette handelt es sich um einen Identifier, welcher der authentifizierten nutzenden Person in dessen Sicherheitskontext und anfragenden Dienst nach der OpenID-Connect-Spezifikation als Subject-Identifier-Typ „pairwise“ zugeordnet ist (siehe Spezifikation OpenID Connect 1.0<sup>17</sup> Kapitel 8.0 „Subject Identifier Type“ und Kapitel 8.1 „Pairwise Identifier Algorithm“ der Spezifikation OpenID Connect 1.0).

Attribut Ebene 0	Typ	Anz.	Bemerkung
<b>iss</b>	String (byte)	1	Der issuer identifier ist eine URL, welche denjenigen identifiziert, der die Inhalte herausgegeben hat.
<b>sub</b>	String (byte)	1	Der subject identifier ist eine Zeichenkette mit einer maximalen Länge von 255 ASCII-Zeichen und repräsentiert, wer lokal nutzt.
<b>aud</b>	String (byte)	1	Die audience (aud) claim listet die erwarteten Empfängerinnen und Empfänger des JWT auf. Im Falle eines ID-Tokens enthält aud die Client-ID der Client-Anwendung, welche den ID-Token angefordert hat.
<b>exp</b>	Integer	1	Das Attribut exp (expires) führt das Datum, zu wann das JWT Gültigkeit hat. Das Datum wird als Unix Epoch-Time

<sup>15</sup> <https://datatracker.ietf.org/doc/html/rfc6749#section-3.3>

<sup>16</sup> <https://www.rfc-editor.org/rfc/rfc7519>

<sup>17</sup> [https://openid.net/specs/openid-connect-core-1\\_0.html](https://openid.net/specs/openid-connect-core-1_0.html)

Attribut Ebene 0	Typ	Anz.	Bemerkung
			geführt und ist ein Integer, welches die Sekunden seit dem 01. Januar 1970 angibt.
<b>iat</b>	Integer	1	Das Attribut iat (issued at) führt das Datum, zu welchem Zeitpunkt das JWT ausgestellt wurde. Das Datum wird als Unix Epoch-Time geführt und ist ein Integer, welches die Sekunden seit dem 01. Januar 1970 angibt.
<b>auth_time</b>	Integer	0..1	Das optionale Attribut auth_time führt das Datum, zu welchem Zeitpunkt die nutzende Person authentifiziert wurde. Das Datum wird als Unix Epoch-Time geführt und ist ein Integer, welches die Sekunden seit dem 01. Januar 1970 angibt.
<b>nonce</b>	String (byte)	0..1	ID-Token können mit dem Attribut nonce ausgestellt werden, um Attacken zu verhindern. Das optionale Attribut wird geführt, sofern dieses bei der Ausstellung angefordert wurde und der SchulConneX-Service diesen entsprechend zum Abgleich vorhält.
<b>acr</b>	String (byte)	0..1	Das optionale Attribut acr (Authentication Context Class Reference) referenziert die URI der Benutzerauthentifizierungsmethode <sup>18</sup> , welche erfüllt wurde.
<b>amr</b>	String (byte)	0..1	Das optionale Attribut amr (Authentication Method Reference) referenziert die Methode der Benutzerauthentifizierung <sup>19</sup> , welche verwendet wurde.
<b>azp</b>	String (byte)	0..1	Das optionale Attribut azp (Authorized Party) führt die Client-ID der autorisierten Anwendung; es handelt sich um die Client-ID des Clients, welche den ID-Token angefordert hat.

Datenmodell 18: ID-Token

## 8.5 Vorbedingungen der Schnittstelle

Generell gilt für alle Anfragen an die API folgende Vorbedingung:

---

<sup>18</sup> ,[https://openid.net/specs/openid-provider-authentication-policy-extension-1\\_0.html](https://openid.net/specs/openid-provider-authentication-policy-extension-1_0.html)

<sup>19</sup> <https://datatracker.ietf.org/doc/html/rfc8176>

Jede API Request muss einen Authorization Header haben. Der Authorization Header setzt sich aus `token_type` und dem eigentlichen `access_token` zusammen. Im Kontext der Spezifikation OpenID Connect muss das Attribut `token_type` den Wert "Bearer" führen.

*Authorization: {token\_type} {access\_token}*

```
Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpJ9.e...
```

*Quellcode 13: Beispiel eines Authorization Headers*

Es wird empfohlen, den zu erwartenden Content-Type explizit als „application/json“ anzugeben. Wird kein Content-Type gesetzt, wird automatisch „application/json“ angenommen.

```
Accept: application/json
```

*Quellcode 14: Beispiel eines Accept Headers*

## 9 API-Endpunkte für Dienstanbieter

### 9.1 Schnittstellenendpunkte für Personen /person-info

Für Dienstanbieter steht derzeit nur ein API-Endpunkt zur Verfügung, welcher das Lesen der Informationen zu einer angemeldeten Person ermöglicht.

#### 9.1.1 Schnittstelle „person-info“

Die Schnittstelle `person-info` bezieht sich auf die Information einer Person.

```
https://subdomain.domain.org/subpath/v1/person-info
```

*Quellcode 15: Beispiel des Schnittstellen-Endpunkts für person-info*

Dieser Endpunkt bezieht sich nur auf die im Sicherheitskontext angemeldete Person und dient somit dem Abruf der Informationen der jeweiligen Person.

Bei Zugriffen auf den Endpunkt ist folgende Operation und HTTP-Methode erlaubt:

Operation	HTTP-Methode
READ	GET

*Tabelle 1: Zugelassene Operation auf dem Endpunkt „person-info“*

#### READ

Ein READ kann nur mit einer HTTP-GET Anfrage erfolgen. Die Antwort erfolgt mit einer pseudonymisierten ID, dem JSON-Objekt der betreffenden Person sowie einer Liste aller für die Sichtbarkeit freigegebenen Personenkontexte.

Bei der Auslieferung von Person-Info wird im HTTP-Header ein ETag mitgeliefert. Bei nachfolgenden Abrufen der Personen-Information vom SchulConneX-Service kann der Dienst ein "If-None-Match" Tag mit diesem ETag in den Request-Header mit aufnehmen.

Hat sich der Datensatz seit Auslieferung des ETag-Headers nicht geändert, so antwortet der Server mit einem Status 304 (Not Modified) und sendet den Datensatz nicht erneut.

```
{
  "pid": <String>, // pseudonymisierte ID
  "person": <Objekt>, // Objekt vom Typ Person, siehe Datenmodell
  "personenkontexte": [ // Array mit Objekten
    <Objekt> // Objekt vom Typ Personenkontext, siehe Datenmodell
  ],
  "gruppen": [
    // Array von Gruppen und Gruppenzugehoerigkeiten eines Personenkontexts
  ]
}
```

```

    "gruppe": ": <Objekt>, // Objekt vom Typ Gruppe
    "gruppenzugehoerigkeiten": {
        <Objekt>
    },
    "sonstige_gruppenzugehoerige":[
        {
            "ktid": <Personenkontext ID>
            "rollen": [<rolle>]
        },
        ....
        {
            "ktid": <Personenkontext ID>
            "rollen": [<rolle>]
        }
    ]
],
"beziehungen"{
    "hat_als_beziehungen": [ // Array von Beziehungen, welche vom
                            Personenkontext ausgehen
    {
        "ktid": <Personenkontext-ID welche der Bezug in
                der angegebenen Rolle ist>
        "beziehung": <Art der Beziehung>
    },
    ...
],
    "ist_von_beziehungen": [
    {
        "ktid": <Personenkontext-ID der den Bezug hat >,
        "beziehung": <Art der Beziehung>
    }
],
}
]

```

```
}
```

Quellcode 16: JSON-Struktur der Rückgabe zu `person-info`

Die pseudonymisierte ID `pid` in der Rückgabe repräsentiert die `id` im aktuell ausgewählten Sicherheitskontext und folgt der gleichen Definition wie die `id`.

Im Sicherheitskontext einer angemeldeten nutzenden Person mit ausschließlichem Zugriff auf den Personenkontext, mit dem sie angemeldet ist, steht nur der ausgewählte und dadurch aktive Personenkontext in der Liste `personenkontexte`. Die `pid` entspricht in diesem Fall der `id` des einen angegebenen Personenkontexts.

Es ist möglich, dass zu einer angemeldeten Person mehrere Personenkontexte bereitgestellt werden. Im Sicherheitskontext einer angemeldeten nutzenden Person mit Zugriff auf mehrere Personenkontexte, die für den Dienst freigegeben sind, umfasst die Liste der Personenkontexte `personenkontexte` alle freigegebenen Personenkontexte. In diesem Fall entspricht die `pid` der Identität der Person und jeder Personenkontext hat die eigene `id`, welche bei Anmeldung im Sicherheitskontext mit ausschließlichem Zugriff auf diesen Personenkontext als `pid` ausgeliefert würde. Die einzelnen Datenmodelle zu „Person“ und „Personenkontext“ sind in 5.3 bzw. 5.4 beschrieben.

Es ist weiterhin möglich, dass eine angemeldete Person mehreren Gruppen zugeordnet ist. In diesem Fall können, geregelt durch eine vertragliche Absprache zwischen Dienstanbieter und Betreiber des SchulConneX-Servers, neben der Gruppenzugehörigkeit im aktuellen Anmeldekontext auch andere Gruppenzugehörigkeiten durch `person-info` geliefert werden.

Abhängig von einer vertraglichen Absprache zwischen Dienstanbieter und Betreiber des SchulConneX-Servers ist es möglich, bei der Gruppenzugehörigkeit gegebenenfalls auch die Personenkontext-IDs und Rollen anderer Mitglieder derselben Gruppe mit auszugeben. Die Personenkontext-IDs und Rollen werden dann in einem Array namens `sonstige_gruppenzugehoerige`, jeweils mit dem Namen `ID` und `Rolle` als Teil der Gruppenzugehörigkeit geliefert,

Typischerweise ist dieses eine Liste von Schüler und Schülerinnen in einer Gruppe, wenn sich eine Lehrkraft mit der Gruppenrolle `Lehr` anmeldet.

Bei der Ausgabe von Beziehungen zwischen Personenkontexten ist es abhängig vom SchulConneX-Server ob alle existierenden und als einsehbar freigegebenen Beziehungen unter „Beziehungen“ mit aufgeführt werden oder nur solche Beziehungen, zu denen der Personenkontext, mit dem eine Beziehung besteht, vorher schon von dem Dienst abgerufen wurde und damit dem Dienst bekannt ist.

```
{  
  "pid": "af3a88fc-d766-11ec-9d64-0242ac120002",
```



```

"person": {
  "referrer": "a141e5ce-7a86-4f92-a600-923d83a5107f",
  "stammorganisation": {
    "id": "fe963bff-e837-4799-91e2-5680222188a5",
    "kennung": "NI_54321",
    "name": "Otto Hahn Schule",
    "anschrift": {
      "postleitzahl": "29614",
      "ort": "Soltau",
      "ortsteil": "Ahlften"
    },
    "typ": "SCHULE"
  },
  "name": {
    "familienname": "Muster",
    "vorname": "Max"
  },
  "geburt": {
    "datum": "2010-01-01",
    "volljaehrig": "ja"
  },
  "geschlecht": "m",
  "lokalisierung": "de",
  "vertrauensstufe": "VOLL"
},
"personenkontexte": [
  {
    "id": "af3a88fc-d766-11ec-9d64-0242ac120002",
    "referrer": "b232e5ce-7a86-4f92-a600-923d83a5208a",
    "organisation": {
      "id": "15685758-d18e-49c1-a644-f9996eb0bf08",
      "kennung": "NI_12345",
      "name": "Muster-Schule",
      "typ": "SCHULE"
    }
  }
]

```

```

    },
    "rolle": "LERN",
    "personenstatus": "AKTIV",
    "gruppen": [
        {
            "gruppe": {
                "id": "b3201d00-f21f-4986-a39d-02a09c8da26c",
                "orgid": "9b3f36ad-9d15-49f9-9660-6cf9746ba446",
                "bezeichnung": "Englischkurs Klasse 3b",
                "typ": "KURS",
                "bereich": "PFLICHT",
                "optionen": [
                    "01"
                ],
                "differenzierung": "G",
                "bildungsziele": [
                    "RS"
                ],
                "jahrgangsstufen": [
                    "03"
                ],
                "faecher": [
                    {
                        "kennung": "EN"
                    }
                ],
                "laufzeit": {
                    "von": "2022-08-01",
                    "bis": "2023-07-31"
                },
            },
        },
        "gruppenzugehoerigkeiten": [
            {
                "id": "c85af1ac-ddaa-4a77-ba74-1836682b9afa",

```

```

            "ktid": "23643c9e-956a-4baa-af4e-
813e0a7efa4b",

            "rollen": [

                "LERN"

            ],

            "von": "2022-08-01",

            "bis": "2023-07-31"

        },

        {

            "id": "b5c17b1c-ee96-4132-9dac-5b165459b972",

            "ktid": "01a51971-6239-4f8a-8973-
e38f5cf21881",

            "rollen": [

                "LEHR"

            ],

            "von": "2022-08-01",

            "bis": "2023-07-31"

        }

    ],

    "sonstige_gruppenzugehoerige": [

        {

            "ktid": " bcdd2717-7eb1-4038-9405-
4fa513650045",

            "rolle": [

                "lern",

                "GMit"

            ]

        },

        {

            "ktid": "a687aa37-b048-46f8-8a61-
64a1ff8e9784",

            "rollen": [

                "lern",

                "GMit"

            ]

        }

    ]

```

```

        }
    ]
}

],
"beziehungen": {
    "hat_als_beziehungen": [
        {
            "ktid": "ade920cc-7ae4-451e-a739-d5f39da7efa0",
            "beziehung": "SchB"
        },
        {
            "ktid": "c757802e-f9e3-40ea-8274-eb404d3a39ce",
            "beziehung": "SchB"
        }
    ],
    "ist_von_beziehungen ": [
        {
            "ktid": "5adb98ef-cdeb-4715-8a48-44a452dd6894",
            "beziehung": "SorgBer"
        }
    ]
}
]
}

```

Quellcode 17: Beispiel einer JSON-Rückgabe zu person-info

#### Mögliche Fehlermeldungen:

Zugrunde liegen die in 7.1.2 definierten Fehlermeldungen, darüber hinaus sind keine spezifischen Fehlermeldungen für diesen Endpunkt zu erwarten.

## 9.2 Person-Info über OIDC Claims

Der bevorzugte Weg, um nach einer Anmeldung auf die Informationen der sich anmeldenden Person zuzugreifen, ist über die API `person-info`.

In einigen Fällen, insbesondere, wenn nur ein kleiner Teil der möglichen Informationen, beispielsweise nur ID, Vorname und Nachname benötigt wird, ist es für Dienstanbieter einfacher, wenn kein gesonderter REST Call auf die API erfolgen muss, sondern die Information direkt als Teil des OIDC ID –Tokens mitgeliefert wird.

Die hier über Claims bereitgestellten Informationen entsprechen den durch `person-info` bereitgestellten Informationen. Wie auch bei den durch die API `person-info` bereitgestellten Informationen, ist dieses im SchulConneX-Server konfigurierbar. Welche Konfiguration vorgenommen wird, ist von der vertraglichen Vereinbarung zwischen dem Dienstanbieter und dem Betreiber des SchulConneX-Servers abhängig.

Es gibt in OIDC eine Reihe von Standard-Claims, wie `sub`, `given_name` oder `family_name`. Wo bereits Standard-Claims existieren, welche den Attributnamen in `person-info` entsprechen, werden die (englischen) Claim-Namen benutzt. Für Attribute, welche kein Äquivalent in den Standard-Claims haben, werden eigene, SchulConneX-spezifische, Claims in URN-Form genutzt.

Die folgende Tabelle gibt das Mapping von Attribut-Namen auf Claim-Namen wieder.

Attribut-Name	Claim-Name
<b>Id</b>	Sub
<b>familiennamen</b>	family_name
<b>vorname</b>	given_name
<b>personenkontext.rolle</b>	urn:schulconnex:de:personenkontext:rolle
<b>personenkontext. organisation. kennung</b>	urn:schulconnex:de:personenkontext:organisation:kennung

*Tabelle 2: Mapping von Attributen auf OIDC-Claim Namen*

Die gelieferten Werte zu den einzelnen Attributen entsprechen **immer** den Konventionen, die auch bei `person-info` genutzt werden und **nicht** den Werten, welche den Standard OIDC Claims entsprechen.

Wird kein Scope Request verwendet, so wird nur der Claim `sub` (entspricht der pseudonymisierten Personen-ID) geliefert.

```
{
  "sub": "af3a88fc-d766-11ec-9d64-0242ac120002"
}
```

Quellcode 18: Beispiel einer JSON-Rückgabe mit OIDC Claims ohne Scope Request

Über einen entsprechenden Scope Request (`scope=person-info`) können auch Vorname, Nachname, Rolle und Organisationskennung mitgeliefert werden.

```
{
  "sub": "af3a88fc-d766-11ec-9d64-0242ac120002",
  "family_name": "Muster",
  "given_name": "Max",
  "urn:schulconnex:de:personenkontext:rolle": "Lern",
  "urn:schulconnex:de:personenkontext:organisation:kennung":
    "NI_12345"
}
```

Quellcode 19: Beispiel einer JSON-Rückgabe mit Scope Requests für `person-info`

### 9.3 Schnittstellenendpunkte für Personen /`personen-info`

Für Dienstanbieter steht dieser API-Endpunkt zur Verfügung, um abzufragen welche Personenkontexte für diesen Dienst noch aktiv sind. Dieses ermöglicht es dem Dienst, persistente Informationen zu anderen Personen und Personenkontexten zu löschen.

Die Schnittstelle sollte regelmäßig aufgerufen werden, um sicherzustellen, dass im SchulConneX-Server gelöschte Daten zeitnah auch bei Dienstanbietern gelöscht werden. Die Häufigkeit des Aufrufes ist mit dem Betreiber des Servers zu vereinbaren.

#### 9.3.1 Schnittstelle „`personen-info`“

Die Schnittstelle `personen-info` liefert die Personenkontext-IDs der dem Dienst bekannten und derzeit aktiven Personenkontexte sowie gegebenenfalls den Löschezitpunkt.

```
https://subdomain.domain.org/subpath/v1/personen-info
```

Quellcode 20: Beispiel des Schnittstellen-Endpunkts für `personen-info`

Der Aufruf dieses Endpunktes erfolgt immer im Kontext des Dienstes. Der Dienst authentisiert sich durch den Client Credentials Flow.

Bei Zugriffen auf den Endpunkt sind folgende Operatione und HTTP-Methode erlaubt:

Operation	HTTP-Methode
-----------	--------------

READ	GET
------	-----

Tabelle 3: Zugelassene Operation auf dem Endpunkt „personen-info“

### READ

Ein READ kann nur mit einer HTTP-GET Anfrage erfolgen. Als Antwort erfolgt eine Liste aus Datenobjekten, welche jeweils die Struktur einer Antwort der API person-info haben, jedoch nur drei Daten-Attribute enthalten: `pid`, `id` und gegebenenfalls den Löschezitpunkt.

Personen und Personenkontexte, welche in der Liste nicht aufgeführt sind, sind als gelöscht anzunehmen und alle lokalen Daten des Dienstes zu dieser Person und diesem Personenkontext sind zu löschen.

Bei der Auslieferung von `person-info` wird im HTTP-Header ein ETag mitgeliefert. Bei nachfolgenden Abrufen der Personen-Information vom SchulConneX-Service kann der Dienst ein "If-None-Match" Tag mit diesem ETag in den Request-Header mit aufnehmen.

Hat sich der Datensatz seit Auslieferung des ETag-Headers nicht geändert, so antwortet der Server mit einem Status 304 (Not Modified) und sendet den Datensatz nicht erneut.

Die Personenkontext-ID `id` ist die für den jeweiligen Dienst individuell pseudonymisierte ID des Personenkontextes.

Die Schnittstelle `personen-info` ermöglicht das Verwenden eines Filterparameters zum Auslesen der vollständigen Datensätze der gelisteten Personen oder Personenkontexte im selben Umfang, wie sie auch bei `person-info` geliefert werden.

Der Filter `vollstaendig` kann die Werte

`personen`, `personenkontexte`, `organisationen`, `gruppen` oder `beziehungen` annehmen. Mehrere Werte können, getrennt durch Kommas, in einem Filter genutzt werden.

Darüber hinaus kann auch eine Filterung nach Personen, Personenkontexten, Gruppen und Organisationen anhand der jeweiligen ID erfolgen.

Welche Filterfunktionen genutzt werden können, wird durch eine vertragliche Absprache zwischen Diensteanbieter und Betreiber des SchulConneX-Servers geregelt.

Filter	Typ	Parameter	Beschreibung
vollstaendig	String	personen	Möglichkeit, statt nur der <code>pid</code> zu jeder der gelisteten Personen die vollständige Beschreibung nach 6.1 Angepasstes Datenmodell Person für Diensteanbieter, anzuzeigen

Filter	Typ	Parameter	Beschreibung
		personenkontexte	Möglichkeit, statt nur <code>id</code> und <code>loeschung</code> zu jedem der gelisteten Personenkontexte die vollständige Beschreibung nach 6.2 Angepasstes Datenmodell Personenkontext für Dienstanbieter, anzuzeigen
		gruppen	Möglichkeit, statt nur der Gruppen-ID zu jeder gelisteten Gruppe die vollständige Beschreibung nach 6.4 Angepasstes Datenmodell Gruppe für Dienstanbieter anzuzeigen
		organisationen	Möglichkeit, statt nur der Organisations-ID zu jeder gelisteten Organisation die vollständige Beschreibung nach 5.2 Organisation anzuzeigen
		beziehungen	Möglichkeit, die Beziehungen, welche aus einem Personenkontext heraus bestehen, aufzulisten  Es werden also nur die ‚hat_als‘ Beziehungen ausgegeben. Der Default ist keine Ausgabe von Beziehungen.
<code>pid</code>	String		Mit diesem Filter kann der Datensatz einer einzelnen Person anhand ihrer <code>pid</code> ausgelesen werden.
<code>personenkontext.id</code>	String		Mit diesem Filter kann der Datensatz einer einzelnen Person mit einem einzelnen Personenkontext anhand der Kontext-ID ausgelesen werden. Die Informationen zur entsprechenden Person werden, unter Berücksichtigung des Filters „vollstaendig“, entsprechend der Client-Berechtigungen ausgeliefert.
<code>gruppe.id</code>	String		Mit diesem Filter können die Datensätze aller Personen ausgegeben werden, welche einer Gruppe zugeordnet sind.



Filter	Typ	Parameter	Beschreibung
			Die Informationen zur entsprechenden Person werden, unter Berücksichtigung des Filters „vollstaendig“, entsprechend der Client-Berechtigungen ausgeliefert.
organisation .id	String		<p>Mit diesem Filter können die Datensätze aller aller Personen ausgegeben werden, welche einer Organisation zugeordnet sind.</p> <p>Die Informationen zur entsprechenden Person werden, unter Berücksichtigung des Filters „vollstaendig“, entsprechend der Client-Berechtigungen ausgeliefert.</p>

Tabelle 4: Zugelassene Filter auf dem Endpunkt /v1/personen-info

```
[
{
  "pid": <String>, // pseudonymisierte ID

  "personenkontexte": [ // Array mit Objekten

  {
    "id": <String>, // ID des Personenkontexts
    "loeschung": {
      "zeitpunkt": <String>, // Loeschzeitpunkt im Format
                        // yyyy-MM-dd'T'hh:mm'Z'
    }
  }
]
}, ...
{
  "pid": <String>, // pseudonymisierte ID
  "personenkontexte": [ // Array mit Objekten
```

```

{
  "id": <String>, // ID des Personenkontexts
  "loeschung": {
    "zeitpunkt": <String>, // Loeschzeitpunkt im Format
                        // yyyy-MM-dd'T'hh:mm'Z'      }
  }
}
]
}]

```

Quellcode 21: JSON-Struktur der Rückgabe zu personen-info

```

[ {
  "pid": "af3a88fc-d766-11ec-9d64-0242ac120002",
  "personenkontexte": [ {
    "id": "16da9a3c-82ac-4bd8-8fa7-8960e6cf8031",
    "loeschung": {
      "zeitpunkt": "2023-01-01T10:00Z"
    }
  }
  ]
}, {
  "pid": "a141e5ce-7a86-4f92-a600-923d83a5107f ",
  "personenkontexte": [ {
    "id": "b8a767cf-a1e2-4a4e-93bc-3eb85b683aeb",
    "loeschung": {
      "zeitpunkt": "2022-10-01T23:59Z"
    }
  }
  ]
}
]

```

Quellcode 22: Beispiel einer JSON-Rückgabe zu personen-info

Beispiel Antwort-Nutzdaten (Response Payload) mit den Filterparametern  
 „?vollstaendig=personen,personenkontexte,organisationen,gruppen“

```

[ {
  "pid": "af3a88fc-d766-11ec-9d64-0242ac120002",
  "änderung_zeit": "2022-12-22T09:30Z",
  "person": {

```

```

"referrer": "a141e5ce-7a86-4f92-a600-923d83a5107f",
"stammorganisation": {
  "id": "fe963bff-e837-4799-91e2-5680222188a5",
  "kennung": "NI_54321"
  "name": "Otto Hahn Schule"
  "anschrift": {
    "postleitzahl": "29614",
    "ort": "Soltau",
    "ortsteil": "Ahlften"
  },
  "typ": "SCHULE"
},
"name": {
  "familienname": "Wolf",
  "vorname": "Sophie"
},
"kommunikation":{
  [{
    "id": "2d5f27e6-e64a-4f04-ac71-75347755c8b3",
    "typ": "EMail",
    "kennung": "Sophie.Wolf@heine-gymnasium-lehrte.de",
  },
  {
    "id": "a2c655c1-b02a-435d-9304-865ba1873f9b",
    "typ": "EMail",
    "kennung": "Sophie.Wolf@gmail.de",
  }
]
}
"geburt": {
  "datum": "2010-07-27",
  "volljaehrig": "ja"
},
"geschlecht": "w",
"lokalisierung": "de-DE",
"vertrauensstufe": "VOLL"
},

```

```

    "personenkontexte": [{
      "id": "af3a88fc-d766-11ec-9d64-0242ac120002",
      "referrer": "b232e5ce-7a86-4f92-a600-923d83a5208a",
      "organisation": {
        "id": "15685758-d18e-49c1-a644-f9996eb0bfaa",
        "kennung": "NI_12345",
        "name": "Heine-Gymnasium",
        "typ": "SCHULE"
      },
      "rolle": "LERN",
      "personenstatus": "AKTIV",
      "gruppen": [{
        "gruppe": {
          "id": "ccc4055b-d1cb-4bb4-b057-
3877d8ad1453",
          "bezeichnung": "Englischkurs Klasse 7b",
          "typ": "KURS",
          "bereich": "PFLICHT",
          "optionen": ["01"],
          "differenzierung": "G",
          "bildungsziele": ["GY-SEK-I"],
          "jahrgangsstufen": ["07"],
          "faecher": [{
            "kennung": "EN"
          }]
        }
      }]
    }],
    {
      "pid": "18c48607-eabe-433a-89ba-77c5d14c4ff0",
      "person": {
        "referrer": "ccfb2742-9207-489e-8cd5-60c3553bae5a",
        "name": {
          "familienname": "Dur",
          "vorname": "Melanie"
        }
      },

```

```

        "geburt": {
            "datum": "2012-03-15",
            "volljaehrig": "ja"
        },
        "geschlecht": "w",
        "lokalisierung": "de-DE",
        "vertrauensstufe": "VOLL"
    },
    "personenkontexte": [{
        "id": "d31ad839-dccf-4db4-96b8-6b8ef3ad727a",
        "referrer": "be49e65e-ff3f-416c-9010-f0a93bb1cbc6",
        "organisation": {
            "id": "15685758-d18e-49c1-a644-f9996eb0bfaa",
            "kennung": "NI_12345",
            "name": "Heine-Gymnasium",
            "typ": "SCHULE"
        },
        "rolle": "LERN",
        "personenstatus": "AKTIV",
        "gruppen": [{
            "gruppe": {
                "id": "ccc4055b-d1cb-4bb4-b057-3877d8ad1453",
                "bezeichnung": "Englischkurs Klasse 7b",
                "typ": "KURS",
                "bereich": "PFLICHT",
                "optionen": ["01"],
                "differenzierung": "G",
                "bildungsziele": ["GY-SEK-I"],
                "jahrgangsstufen": ["07"],
                "faecher": [{
                    "kennung": "EN"
                }]
            }
        }]
    }]
},

```

```

{
  "pid": "6c3b40ae-c5cf-4257-92d4-d48d80a7cc0e",
  "änderung_zeit": "2022-12-17T12:15Z",
  "person": {
    "referrer": "2640clea-e272-4932-a560-0d08295b78ec",
    "name": {
      "familienname": "Lehmann",
      "vorname": "Kevin"
    },
    "geburt": {
      "datum": "2013-11-25",
      "volljaehrig": "nein"
    },
    "geschlecht": "m",
    "lokalisierung": "de-DE",
    "vertrauensstufe": "VOLL"
  },
  "personenkontexte": [{
    "id": "d4c66222-fec0-4b39-a02e-97b41a77365e",
    "referrer": "8d8ba096-916c-41ba-b852-5d4d86f4926f",
    "organisation": {
      "id": "15685758-d18e-49c1-a644-f9996eb0bfaa",
      "kennung": "NI_12345",
      "name": "Heine-Gymnasium",
      "typ": "SCHULE"
    },
    "rolle": "LERN",
    "personenstatus": "AKTIV",
    "gruppen": [{
      "gruppe": {
        "id": "68f6ed9b-f00a-460f-80ec-0aa081da30aa",
        "bezeichnung": "Englischkurs Klasse 1",
        "typ": "KURS",
        "bereich": "PFLICHT",
        "optionen": ["01"],
        "differenzierung": "G",
        "bildungsziele": ["GY-SEK-I"],

```

```

        "jahrgangsstufen": ["01"],
        "faecher": [{
            "kennung": "EN"
        }]
    }
}
}
}
}
]

```

Quellcode 23: Beispiel einer JSON-Rückgabe zu personen-info mit Filterparametern 'vollstaendig'

Beispiel Antwort-Nutzdaten (Response Payload) mit den Filterparametern „?vollstaendig=personen,personenkontexte&gruppe.id=ccc4055b-d1cb-4bb4-b057-3877d8ad1453“ zur Auswahl aller Personen in einer bestimmten Gruppe, mit vollen Personen und Personenkontext-Informationen, aber nur den IDs von Gruppe und Organisation

```

[ {
    "pid": "af3a88fc-d766-11ec-9d64-0242ac120002",
    "person": {
        "referrer": "a141e5ce-7a86-4f92-a600-923d83a5107f",
        "name": {
            "familienname": "Wolf",
            "vorname": "Sophie"
        },
        "geburt": {
            "datum": "2010-07-27",
            "volljaehrig": "ja"
        },
        "geschlecht": "w",
        "lokalisierung": "de-DE",
        "vertrauensstufe": "VOLL"
    },
    "personenkontexte": [{
        "id": "af3a88fc-d766-11ec-9d64-0242ac120002",
        "referrer": "b232e5ce-7a86-4f92-a600-923d83a5208a",
        "organisation": {
            "id": "15685758-d18e-49c1-a644-f9996eb0bfaa"
        }
    }]
} ]

```

```

    },
    "rolle": "LERN",
    "personenstatus": "AKTIV",
    "gruppen": [{
        "gruppe": {
            "id": "ccc4055b-d1cb-4bb4-b057-3877d8ad1453"
        }
    }]
}],
{
    "pid": "18c48607-eabe-433a-89ba-77c5d14c4ff0",
    "person": {
        "referrer": "ccfb2742-9207-489e-8cd5-60c3553bae5a",
        "name": {
            "familienname": "Dur",
            "vorname": "Melanie"
        },
        "geburt": {
            "datum": "2012-03-15",
            "volljaehrig": "ja"
        },
        "geschlecht": "w",
        "lokalisierung": "de-DE",
        "vertrauensstufe": "VOLL"
    },
    "personenkontexte": [{
        "id": "d31ad839-dccf-4db4-96b8-6b8ef3ad727a",
        "referrer": "be49e65e-ff3f-416c-9010-f0a93bb1cbc6",
        "organisation": {
            "id": "15685758-d18e-49c1-a644-f9996eb0bfaa"
        },
        "rolle": "LERN",
        "personenstatus": "AKTIV",
        "gruppen": [{
            "gruppe": {
                "id": "ccc4055b-d1cb-4bb4-b057-3877d8ad1453"
            }
        }]
    }]
}

```



```
    }  
  }  
}  
]
```

*Quellcode 24: Beispiel einer JSON-Rückgabe zu personen-info mit Filterparametern 'vollstaendig' und Auswahl nach Gruppe*

#### *Mögliche Fehlermeldungen:*

Zugrunde liegen die in 7.1.2 definierten Fehlermeldungen, darüber hinaus sind keine spezifischen Fehlermeldungen für diesen Endpunkt zu erwarten.

## 10 API-Endpunkte für Quellsysteme

### 10.1 Schnittstellen für Personen

#### 10.1.1 [/personen] Personendatensätze erstellen und auflisten

Die Schnittstelle /v1/personen bezieht sich auf das Erstellen eines Personendatensatzes und die Auflistung von Personendatensätzen.

<code>https://subdomain.domain.org/subpaths/v1/personen</code>
--

Quellcode 25: URL-Struktur des Endpunkts personen

Bei Zugriffen auf den Endpunkt sind folgende Operationen und HTTP-Methoden erlaubt:

Operation	HTTP-Methode
CREATE	POST
READ	GET

Tabelle 5: Zugelassene Operationen auf dem Endpunkt /v1/personen

Die Schnittstelle /v1/personen ermöglicht das Verwenden von Filterparametern zur Präzisierung der Anfrage bei HTTP-GET. Werden mehrere Filter angegeben, so sind sie als mit 'und' verknüpft anzusehen. Die mehrfache Nutzung des gleichen Filters in einer Anfrage ist nicht zulässig. Folgende Tabelle listet die zur Verfügung stehenden Filter:

Filter	Typ	Beschreibung
referrer	String	Möglichkeit, die Ergebnisliste der Anfrage nach dem Fremdschlüssel "referrer" zu filtern  Der SchulConneX-Server überprüft, ob der Wert des Filterparameters ohne Beachtung von Groß- und Kleinschreibung im Filter-Attribut beinhaltet ist (contains).
familiename	String	Möglichkeit, die Ergebnisliste der Anfrage nach dem Attribut "person.name.familiename" zu filtern  Der SchulConneX-Server überprüft, ob der Wert des Filterparameters ohne Beachtung von Groß- und Kleinschreibung im Filter-Attribut beinhaltet ist (contains).

Filter	Typ	Beschreibung
vorname	String	<p>Möglichkeit, die Ergebnisliste der Anfrage nach dem Attribut "person.name.vorname" zu filtern</p> <p>Der SchulConneX-Server überprüft, ob der Wert des Filter-parameters ohne Beachtung von Groß- und Kleinschreibung im Filter-Attribut beinhaltet ist (contains).</p>
sichtfreigabe	String	<p>erlaubt es Personen, abhängig davon aufzulisten, ob diese von einer anderen Organisation zum Lesen freigegeben wurden</p> <p>Die möglichen Optionen sind "sichtfreigabe=ja", um nur die durch externe Organisationen freigegebenen Personen beziehungsweise Personenkontexte aufzulisten und "sichtfreigabe=nein", um nur eigene Personen beziehungsweise Personenkontexte aufzulisten. Wird nicht nach Sichtfreigabe gefiltert, so werden alle lesbaren Personen, beziehungsweise Personenkontexte geliefert.</p>

*Tabelle 6: Zugelassene Filter auf dem Endpunkt /v1/personen*

## CREATE

Mittels dieser Schnittstelle wird ein neuer Datensatz "Person" erstellt.

Ein CREATE ist nur mit HTTP-POST auf die API /v1/personen zum Erstellen von Personendatensätzen zugelassen. Die Anfrage-Nutzdaten (Request Payload) beinhalten ein JSON-Objekt des Datentyps Personendatensatz. Das Datenmodell „Personendatensatz“ ist in 5.5 beschrieben. Das Datenmodell „Person“ ist in 5.3 beschrieben.

### Request

Die folgende Tabelle listet die Attribute einer Person ("personendatensatz.person"), welche von einem Quellsystem oder Dienst nicht gesetzt werden können.

Attribut	In den Anfrage-Nutzdaten erforderlich?	Bemerkung
id	nein	ID der Person wird vom SchulConneX-Server vergeben und ist eindeutig Dieses Attribut ist unveränderbar (immutable).
mandant	nein	ID des Mandanten, dem die Personen zugeordnet ist wird vom SchulConneX-Server vergeben und ist eindeutig Dieser Wert ist eine Referenz auf einen Mandanten.
revision	nein	Revision der Personen wird vom SchulConneX-Server mit der Erstellung des Datensatzes sowie Aktualisierung generiert Dieser Wert kann nicht von Quellsystemen oder Diensten gesetzt werden.

*Tabelle 7: Liste von Attributen, welche durch ein Quellsystem oder Dienst über den Endpunkt /v1/personen auf HTTP-POST nicht gesetzt werden können.*

In Anbetracht der Attribute, die von einem Quellsystem oder Dienst bei der Erstellung einer Person nicht gesetzt oder verändert werden können, ergibt sich folgende JSON-Struktur der Anfrage-Nutzdaten (Request Payload):

```

{
  "referrer": <String>, // ID der Person im Quellsystem.
  "name": {
    "familienname": <String>,
    //Familienname der Person. Dieses Feld ist ein Pflichtfeld.
    "vorname": <String>,
    // Vorname der Person. Dieses Feld ist ein Pflichtfeld.
    "initialenfamilienname": <String>,
    //Initial oder Initialen des Familiennames der Person. D
    "initialenvorname ": <String>,
    // Initial oder Initialen des Vornamens der Person.
    "rufname": <String>, // Rufname der Person.
    "titel": <String>, // Titel der Person.
    "anrede": [
      <String> // Liste von Anreden der Person.
    ],
    "namenssuffix": [
      <String> // Liste von Namenssuffixe der Person.
    ],
    "sortierindex": <String> // sortierindex des Nachnamens
  },
  "geburt": {
    "datum": <Datum>, // Geburtsdatum der Person.
    "geburtsort": <String> // Geburtsort der Person.
  },
  "geschlecht": <Code>,
  // Geschlecht der Person. Siehe Code-Liste "Geschlecht".
  "lokalisierung": <Code>, // Lokalisierung. Siehe RFC5646.
  "vertrauensstufe": <Code>,
  // Vertrauensstufe. Siehe Code-Liste "Vertrauensstufe".
  "auskunftssperre": <Code>
  // Auskunftssperre. Siehe Code-Liste "Boolean".
  // Dieses Feld ist ein Pflichtfeld.
}

```

*Quellcode 26: JSON-Struktur eines CREATE-Requests an /v1/personen*

#### Beispiel Anfrage-Nutzdaten (Request Payload):

```

{
  "referrer" : "125",

```

```

    "name" :

    {

        "familienname" : "von Musterfrau",

        "vorname" : "Natalie",

        "initialenfamilienname" : "M",

        "initialenvorname " : "N",

        "sortierindex": "4"

    },

    "geburt" :

    {

        "datum" : "2005-05-01",

        "geburtsort" : "Berlin, Deutschland"

    },

    "geschlecht" : "w",

    "lokalisierung" : "de-DE",

    "vertrauensstufe" : "VOLL"

}

```

*Quellcode 27: Beispiel Request zu /personen auf HTTP-POST*

## Response

Bei einer erfolgreichen Anforderung zum Erstellen eines Personendatensatzes wird diese Anforderung mit einer Repräsentation des Personendatensatzes in den Antwort-Nutzdaten (Response Payload) und dem HTTP Status Code 200 („200 OK“) quittiert.

Nachfolgend die JSON-Struktur der Antwort-Nutzdaten (Response Payload):

```
<Person> // Siehe "Person"
```

*Quellcode 28: JSON-Struktur einer CREATE-Response an /v1/personen*

Beispiel Antwort-Nutzdaten (Response Payload):

```

{

    "id" : "a6e1a860-8d44-4b2b-aef7-aa2c8bf5beb5",

}

```

```

    "referrer" : "125",
    "mandant" : "58f45270-8e54-40c6-a212-980307fc19be",
    "name" :
    {
        "familienname" : "von Musterfrau",
        "vorname" : "Natalie",
        "initialenfamilienname" : "M",
        "initialenvorname" : "N",
        "sortierindex": "4"
    },
    "geburt" :
    {
        "datum" : "2005-05-01",
        "geburtsort" : "Berlin, Deutschland"
    },
    "geschlecht" : "w",
    "lokalisierung" : "de-DE",
    "vertrauensstufe" : "VOLL",
    "revision" : "1"
}

```

*Quellcode 29: Beispiel Response zu /personen auf HTTP-POST*

### Mögliche Fehlermeldungen

Zugrunde liegen die in 7.1.2 definierten Fehlermeldungen.

Darüber hinaus gibt es keine spezifischen Fehlermeldungen für diesen Endpunkt.

## READ

Mittels dieses Endpunktes werden alle Personendatensätze zurückgegeben, auf die der Quellsystemanbieter zugreifen darf.

Ein READ kann nur mit einer HTTP-GET Anfrage erfolgen.

### Request

Die Anfrage erfordert keine Nutzdaten (Request Payload).

### Response

Die Antwort-Nutzdaten (Response Payload) beinhalten ein Array von JSON-Objekten vom Datentyp Personendatensatz, sofern Einträge (ggf. passend zu den optionalen Filtern) existieren. Das Datenmodell „Personendatensatz“ ist in 5.5 beschrieben. Das Datenmodell „Person“ ist in 5.3 beschrieben.

Die Ergebnisliste beinhaltet ausschließlich Datensätze vom Datentyp Person bzw. erweitert über das Attribut „personendatensatz.personenkontexte“ Personenkontexte, welche der eigenen Organisation zugeordnet werden können. Der SchulConneX-Server übernimmt diese Information aus dem access\_token.

Nachfolgend die JSON-Struktur der Antwort-Nutzdaten (Response Payload):

```
[
  {
    "person": <Person>, // Siehe "Person".
    "personenkontexte": [
      <Personenkontext> // Siehe "Personenkontext".
    ]
  }
]
```

*Quellcode 30: JSON-Struktur eines READ-Requests an /v1/personen*

Beispiel Antwort-Nutzdaten (Response Payload):

```
[ {
  "person": {
    "id": "b0d7b0dd-3477-4122-a38d-095ec242e788",
    "referrer": "123",
```



```

        "mandant": "58f45270-8e54-40c6-a212-980307fc19be",
        "name": {
            "familienname": "Muster",
            "vorname": "Max",
            "initialenfamilienname": "M",
            "initialenvorname": "M",
            "sortierindex": "0"
        },
        "geburt": {
            "datum": "2000-01-01",
            "geburtsort": "Berlin, Deutschland"
        },
        "geschlecht": "d",
        "lokalisierung": "de-DE",
        "vertrauensstufe": "VOLL",
        "revision": "1"
    },
    "personenkontexte": [{
        "id": "4d0f579c-0b9a-4d3a-b484-87b3bee8a2ad",
        "referrer": "PeKt_54321",
        "mandant": "58f45270-8e54-40c6-a212-980307fc19be",
        "organisation": {
            "id": "c7de2c10-26a6-4d12-8152-f9e8db497625"
        },
        "rolle": "LERN",
        "personenstatus": "AKTIV",
        "revision": "1"
    },
    {
        "person": {
            "id": "3d41fe34-e6fe-4777-a2ae-95b9823209a1",

```

```

        "referrer": "124",
        "mandant": "58f45270-8e54-40c6-a212-980307fc19be",
        "name": {
            "familienname": "Doe",
            "vorname": "Jane",
            "initialenfamilienname": "D",
            "initialenvorname": "J",
            "sortierindex": "0"
        },
        "geburt": {
            "datum": "2004-04-01",
            "geburtsort": "Berlin, Deutschland"
        },
        "geschlecht": "w",
        "lokalisierung": "de-DE",
        "vertrauensstufe": "VOLL",
        "revision": "1"
    }
}

```

*Quellcode 31: Beispiel Response zu /personen auf HTTP-GET*

Durch das Setzen von Filterparametern (query-parameter) wird mit der HTTP-GET-Anfrage eine Filter-Operation mitgeschickt, welche die Ergebnismenge der Antwort-Nutzdaten (Response Payload) beeinflusst.

Beispiel Antwort-Nutzdaten (Response Payload) mit den Filterparametern „?referrer=123“:

```

[ {
    "person": {
        "id": "b0d7b0dd-3477-4122-a38d-095ec242e788",
        "referrer": "123",
        "mandant": "58f45270-8e54-40c6-a212-980307fc19be",
        "name": {

```

```

        "familienname": "Muster",
        "vorname": "Max",
        "initialenfamilienname": "M",
        "initialenvorname ": "M",
        "sortierindex": "0"
    },
    "geburt": {
        "datum": "2000-01-01",
        "geburtsort": "Berlin, Deutschland"
    },
    "geschlecht": "d",
    "lokalisierung": "de-DE",
    "vertrauensstufe": "VOLL",
    "revision": "1"
},
"personenkontexte": [{
    "id": "4d0f579c-0b9a-4d3a-b484-87b3bee8a2ad",
    "referrer": "PeKt_54321",
    "mandant": "58f45270-8e54-40c6-a212-980307fc19be",
    "organisation": {
        "id": "c7de2c10-26a6-4d12-8152-f9e8db497625"
    },
    "rolle": "LERN",
    "personenstatus": "AKTIV",
    "revision": "1"
}]
}]

```

Quellcode 32: Beispiel Response zu /personen auf HTTP-GET mit den Filter-parametern (query-parameter) „?referrer=123“

### Mögliche Fehlermeldungen

Zugrunde liegen die in 7.1.2 definierten Fehlermeldungen.

Darüber hinaus gibt es keine spezifischen Fehlermeldungen für diesen Endpunkt.

### 10.1.2 [/personen/{id}] Individuelle Personendatensätze lesen, ändern und löschen

Die Schnittstelle /v1/personen/{personendatensatz.person.id} bezieht sich auf das Anfordern, Bearbeiten und Löschen eines Personendatensatzes per ID.

```
https://subdomain.domain.org/subpaths/v1/personen/{personendatensatz.person.id}
```

*Quellcode 33: URL-Struktur des Endpunkts personen/{personendatensatz.person.id}*

Der Pfad-Parameter {personendatensatz.person.id} bezieht sich auf die vom Quellsystem vergebene ID der Person.

Bei Zugriffen auf den Endpunkt sind folgende Operationen und HTTP-Methoden erlaubt:

Operation	HTTP-Methode
READ	GET
UPDATE	PUT
DELETE	DELETE

*Tabelle 8: Zugelassene Operationen auf dem Endpunkt /v1/personen/{personendatensatz.person.id}*

## READ

Dieser Schnittstellenendpunkt liefert den Personendatensatz zur angeforderten ID.

Ein READ ist nur mit HTTP-GET auf die API `/v1/personen/{personendatensatz.person.id}` zum Abfragen eines Personendatensatzes per ID zugelassen. Die Antwort-Nutzdaten (Response Payload) beinhalten ein JSON-Objekt des Datentyps Personendatensatz. Das Datenmodell „Personendatensatz“ ist in 5.5 beschrieben. Das Datenmodell „Person“ ist in 5.3 beschrieben.

### Request

Die Anfrage erfordert keine Nutzdaten (Request Payload).

### Response

Nachfolgend die JSON-Struktur der Antwort-Nutzdaten (Response Payload):

```
{
  "person": <Person>, // Siehe "Person".
  "personenkontexte": [
    <Personenkontext> // Siehe "Personenkontext".
  ]
}
```

*Quellcode 34: JSON-Struktur der Antwort bei READ auf `/v1/personen/{personendatensatz.person.id}`*

Beispiel Antwort-Nutzdaten (Response Payload):

```
{
  "person": {
    "id": "a6e1a860-8d44-4b2b-aef7-aa2c8bf5beb5",
    "referrer": "125",
    "mandant": "58f45270-8e54-40c6-a212-980307fc19be",
    "name": {
      "familienname": "von Musterfrau",
      "vorname": "Natalie",
      "initialenfamilienname": "M",
      "initialenvorname": "N",
      "sortierindex": "4"
    },
  },
}
```

```

    "geburt": {
        "datum": "2005-05-01",
        "geburtsort": "Berlin, Deutschland"
    },
    "geschlecht": "w",
    "lokalisierung": "de-DE",
    "vertrauensstufe": "VOLL",
    "revision": "1"
},
"personenkontexte": [{
    "id": "4d0f579c-0b9a-4d3a-b484-87b3bee8a2ad",
    "referrer": "PeKt_54321",
    "mandant": "58f45270-8e54-40c6-a212-980307fc19be",
    "organisation": {
        "id": "c7de2c10-26a6-4d12-8152-f9e8db497625"
    },
    "rolle": "LERN",
    "personenstatus": "AKTIV",
    "revision": "1"
}]
}

```

*Quellcode 35: Beispiel Response zu /personen/{personendatensatz.person.id} auf HTTP-GET*

### Mögliche Fehlermeldungen

Zugrunde liegen die in 7.1.2 definierten Fehlermeldungen.

Darüber hinaus gibt es keine spezifischen Fehlermeldungen für diesen Endpunkt.

## UPDATE

Dieser Schnittstellenendpunkt aktualisiert den Personendatensatz der angegebenen ID.

Ein UPDATE ist nur mit HTTP-PUT auf die API `/v1/personen/{personendatensatz.person.id}` zum Aktualisieren eines Personendatensatzes per ID zugelassen. Die Anfrage-Nutzdaten (Request Payload) beinhalten ein JSON-Objekt des Datentyps Personendatensatz. Das Datenmodell „Personendatensatz“ ist in 5.5 beschrieben. Das Datenmodell „Person“ ist in 5.3 beschrieben.

Es ist erforderlich, dass zur Aktualisierungsanfrage im bearbeiteten Personendatensatz das Attribut „personendatensatz.person.revision“ des zugrunde liegenden Personendatensatzes mitgeschickt wird. Der SchulConneX-Service überprüft anhand des mitgeschickten Wertes des Attributs „personendatensatz.person.revision“, ob der Datensatz der Person in der Zwischenzeit keine Änderung erfahren hat. Der Wert des Attributs kann von einem Quellsystem oder Dienst nicht verändert werden; dieser wird mit der Erstellung (Create) des Datensatzes Person und jeder erfolgreichen Änderung (Update) vom SchulConneX-Server neu generiert.

Die folgende Tabelle listet die Attribute einer Person („personendatensatz.person“), welche von einem Quellsystem oder Dienst nicht verändert werden können.

Attribut	In den Anfrage Nutzdaten erforderlich?	Bemerkung
id	nein	ID der Person wird vom SchulConneX-Server vergeben und ist eindeutig Dieses Attribut ist unveränderbar (immutable).
mandant	Nein	ID des Mandanten, dem die Personen zugeordnet ist wird vom SchulConneX Server vergeben und ist eindeutig Dieser Wert ist eine Referenz auf einen Mandanten.
kommunikation	nein	wird über API <code>/erreichbarkeiten/{erreichbarkeit.id}</code> gesetzt oder entfernt
revision	Ja	Revision des Datensatzes

		<p>wird vom SchulConneX-Server mit der Erstellung des Datensatzes sowie Aktualisierung generiert</p> <p>Dieser Wert kann nicht von Quellsystemen oder Diensten gesetzt werden.</p> <p>Bei einem Update eines Datensatzes muss der letzte Revisionswert für diesen Datensatz mitgegeben werden, damit erkannt werden kann, ob der Datensatz inzwischen anderweitig modifiziert wurde.</p>
--	--	--

*Tabelle 9: Liste von Attributen, welche durch ein Quellsystem oder Dienst über den Endpunkt /v1/personen/{personendatensatz.person.id} auf HTTP-PUT nicht verändert werden können.*

## Request

Nachfolgend die JSON-Struktur der Anfrage-Nutzdaten (Request Payload):

```
{
  <Person> // Siehe "Person".
}
```

*Quellcode 36: JSON-Struktur eines READ-Requests an /v1/personen/{personendatensatz.person.id}*

In Anbetracht der Attribute, die von einem Quellsystem oder Dienst bei der Aktualisierung einer Person nicht gesetzt oder verändert werden können, ergibt sich folgende JSON-Struktur der Anfrage-Nutzdaten (Request Payload):

```
{
  "referrer": <String>, // ID der Person im Quellsystem.
  "name": {
    "familienname": <String>,
    //Familienname der Person. Dieses Feld ist ein Pflichtfeld.
    "vorname": <String>,
    // Vorname der Person. Dieses Feld ist ein Pflichtfeld.
    "initialenfamilienname": <String>,
    //Initial oder Initialen des Familiennames der Person. D
    "initialenvorname ": <String>,
    // Initial oder Initialen des Vornamens der Person.
    "rufname": <String>,
    // Rufname der Person.
    "titel": <String>, // Titel der Person.
    "anrede": [
      <String> // Liste von Anreden der Person.
    ]
  }
}
```



```

    ],
    "namenssuffix": [
        <String> // Liste von Namenssuffixe der Person.
    ],
    "sortierindex: <String> // Sortierindex des Nachnamens
},
"geburt": {
    "datum": <Datum>, // Geburtsdatum der Person.
    "geburtsort": <String> // Geburtsort der Person.
},
"geschlecht": <Code>,
// Geschlecht der Person. Siehe Code-Liste „Geschlecht“.
"lokalisierung": <Code>, // Lokalisierung. Siehe RFC5646.
"vertrauensstufe": <Code>,
// Vertrauensstufe. Siehe Code-Liste „Vertrauensstufe“.
"auskunftssperre": <Code>,
// Auskunftssperre. Siehe Code-Liste „Boolean“.
// Dieses Feld ist ein Pflichtfeld.
"revision": <String>
// Die Revision der Person des
// Personendatensatzes. Dieses Feld ist ein Pflichtfeld.
}

```

*Quellcode 37: Spezielle JSON-Struktur eines READ-Requests an /v1/personen/{personendatensatz.person.id}*

### Beispiel Anfrage-Nutzdaten (Request Payload):

```

{
    "referrer": "125",
    "name": {
        "familienname": "von Musterfrau",
        "vorname": "Natalie",
        "initialenfamilienname": "M",
        "initialenvorname ": "N",
        "sortierindex": "4"
    },
    "geburt": {
        "datum": "2005-05-01",

```

```

        "geburtsort": "Berlin, Deutschland"

    },

    "geschlecht": "w",

    "lokalisierung": "de-DE",

    "vertrauensstufe": "VOLL",

    "revision": "1"

}

```

*Quellcode 38: Beispiel Request zu /personen/{personendatensatz.person.id} auf HTTP-PUT*

Anmerkung: Bei dieser Aktualisierungsanfrage handelt es sich um eine HTTP-PUT-Operation, bei der die Anfrage-Nutzlast mit den gegebenen Attributen der Aktualisierung zugrunde gelegt wird (mitgeschickte Attribute werden aktualisiert, nicht mitgeschickte Attribute werden gelöscht); es ist keine HTTP-PATCH-Operation, bei der lediglich die zu aktualisierenden Attribute in der Anfrage-Nutzlast mitgeschickt werden.

### Response

Bei einer erfolgreichen Anforderung zum Aktualisieren eines Personendatensatzes wird diese Anforderung mit einer Repräsentation des Personendatensatzes in den Antwort-Nutzdaten (Response Payload) und dem HTTP Status Code 200 („200 OK“) quittiert.

Nachfolgend die JSON-Struktur der Antwort-Nutzdaten (Response Payload):

```

{
    <Person> // Siehe "Person".
}

```

*Quellcode 39: Beispiel Request zu /personen/{personendatensatz.person.id} auf HTTP-PUT*

Beispiel Antwort-Nutzdaten (Response Payload):

```

{

    "id": "a6e1a860-8d44-4b2b-aef7-aa2c8bf5beb5",

    "referrer": "125",

    "mandant": "58f45270-8e54-40c6-a212-980307fc19be",

    "name": {

        "familienname": "von Musterfrau",

        "vorname": "Natalie",

        "initialenfamilienname": "M",

    }

}

```

```
        "initialenvorname ": "N",
        "sortierindex": "4"
    },
    "geburt": {
        "datum": "2005-05-01",
        "geburtsort": "Berlin, Deutschland"
    },
    "geschlecht": "w",
    "lokalisierung": "de-DE",
    "vertrauensstufe": "VOLL",
    "revision": "2"
}
```

*Quellcode 40: Beispiel Response zu /personen/{personendatensatz.person.id} auf HTTP-PUT*

### Mögliche Fehlermeldungen

Zugrunde liegen die in 7.1.2 definierten Fehlermeldungen.

Darüber hinaus gibt es keine spezifischen Fehlermeldungen für diesen Endpunkt.

## DELETE

Dieser Schnittstellenendpunkt löscht den Personendatensatz zur angeforderten ID.

Ein DELETE ist nur mit HTTP-DELETE auf die API `/v1/personen/{personendatensatz.person.id}` zum Löschen eines Personendatensatzes per ID zugelassen. Die Anfrage-Nutzdaten (Request Payload) beinhalten ein JSON-Objekt des Datentyps Personendatensatz. Das Datenmodell „Personendatensatz“ ist in 5.5 beschrieben. Das Datenmodell „Person“ ist in 5.3 beschrieben.

Es ist erforderlich, dass für eine Löschanfrage eines Personendatensatzes das Attribut „personendatensatz.person.revision“ des zugrunde liegenden Personendatensatzes mitgeschickt wird. Der SchulConneX-Server überprüft anhand des mitgeschickten Wertes des Attributs „personendatensatz.person.revision“, ob der Datensatz der Person in der Zwischenzeit keine Änderung erfahren hat.

### Request

Nachfolgend die JSON-Struktur der Anfrage Nutzdaten (Request Payload):

```
{
  "revision": <String>
  // Revision der Person des zugrunde liegenden Personendatensatzes.
  // Dieses Feld ist ein Pflichtfeld.
}
```

*Quellcode 41: JSON-Struktur eines DELETE-Requests an `/v1/personen/{personendatensatz.person.id}`*

Beispiel Anfrage Nutzdaten (Request Payload):

```
{
  "revision": "2"
}
```

*Quellcode 42: Beispiel Request zu `/personen/{personendatensatz.person.id}` auf HTTP-DELETE*

### Response

Bei einer erfolgreichen Ausführung der Löschanfrage wird es keine Antwort-Nutzdaten (Response Payload) geben. Eine erfolgreiche Löschanforderung eines Personendatensatz per ID wird mit dem HTTP Status Code 204 („204 No Content“) quittiert.

### Mögliche Fehlermeldungen

Zugrunde liegen die in 7.1.2 definierten Fehlermeldungen.

Ein Fehler, der nur bei dieser API auftreten kann, ist Error 400, Subcode 12: „Person enthält noch Personenkontexte.“ Dieser tritt dann auf, wenn versucht wird, den Datensatz einer Person zu löschen, für die noch Personenkontexte bestehen.

Darüber hinaus gibt es keine spezifischen Fehlermeldungen für diesen Endpunkt.

### 10.1.3 [/personen/{id}/personenkontexte] Personenkontexte lesen und erstellen

Die Schnittstelle /v1/personen/{personendatensatz.person.id}/personenkontexte bezieht sich auf das Erstellen eines Personenkontexts und die Auflistung von Personenkontexten zu einem bestimmten Personendatensatz.

```
https://subdomain.domain.org/subpaths/v1/personen/{personendatensatz.person.id}/personenkontexte
```

Quellcode 43: URL-Struktur des Endpunkts personen/{personendatensatz.person.id}/personenkontexte

Der Pfad-Parameter {personendatensatz.person.id} bezieht sich auf die vom Quellsystem vergebene ID der Person.

Bei Zugriffen auf den Endpunkt sind folgende Operationen und HTTP-Methoden erlaubt:

Operation	HTTP-Methode
CREATE	POST
READ	GET

Tabelle 10: Zugelassene Operationen auf dem Endpunkt /v1/personen/{personendatensatz.person.id}/personenkontexte

Die Schnittstelle /v1/personen/{personendatensatz.person.id}/personenkontexte ermöglicht das Verwenden von Filterparametern zur Präzisierung der Anfrage bei HTTP-GET. Werden mehrere Filter angegeben, so sind sie als mit 'und' verknüpft anzusehen. Die mehrfache Nutzung des gleichen Filters in einer Anfrage ist nicht zulässig. Folgende Tabelle listet die zur Verfügung stehenden Filter:

Filter	Typ	Beschreibung
referrer	String	Möglichkeit, die Ergebnisliste der Anfrage nach dem Fremdschlüssel "referrer" zu filtern  Der SchulConneX-Server überprüft, ob der Wert des Filterparameters ohne Beachtung von Groß- und Kleinschreibung im Filter-Attribut beinhaltet ist (contains).
rolle	String (Code)	Möglichkeit, die Ergebnisliste der Anfrage nach dem Attribut "personenkontext.rolle" zu filtern

Filter	Typ	Beschreibung
		<p>Der SchulConneX-Server überprüft, ob der Wert des Filterparameters mit dem Wert des Filter-Attributs übereinstimmt (equals).</p> <p>Dem Filterparameter liegt eine Codeliste C3 (Rolle) zugrunde.</p>
personenstatus	String (Code)	<p>Möglichkeit, die Ergebnisliste der Anfrage nach dem Attribut "personenkontext.personenstatus" zu filtern</p> <p>Der SchulConneX-Server überprüft, ob der Wert des Filterparameters mit dem Wert des Filter-Attributs übereinstimmt (equals).</p> <p>Dem Filterparameter liegt eine Codeliste Personenstatus zugrunde.</p>
sichtfreigabe	String	<p>erlaubt es Personen, abhängig davon aufzulisten, ob diese von einer anderen Organisation zum Lesen freigegeben wurden</p> <p>Die möglichen Optionen sind "sichtfreigabe=ja", um nur die durch externe Organisationen freigegebenen Personen beziehungsweise Personenkontexte aufzulisten, und "sichtfreigabe=nein", um nur eigene Personen beziehungsweise Personenkontexte aufzulisten. Wird nicht nach Sichtfreigabe gefiltert, so werden alle lesbaren Personen, beziehungsweise Personenkontexte geliefert.</p>

Tabelle 11: Zugelassene Filter auf dem Endpunkt `/v1/personen/{personendatensatz.person.id}/personenkontexte`

## CREATE

Dieser Schnittstellenendpunkt erstellt einen Personenkontext zur angegebenen Person per ID {personendatensatz.person.id}.

Ein CREATE ist nur mit HTTP-POST auf die API /v1/personen/{personendatensatz.person.id}/personenkontexte zum Erstellen von Personenkontexten zu einem Personendatensatz zugelassen. Die Anfrage-Nutzdaten (Request Payload) beinhalten ein JSON-Objekt des Datentyps Personenkontext. Das Datenmodell „Personenkontext“ ist in 5.4 beschrieben.

Beim Erstellen eines Personenkontexts ist der Default-Status des Attributes “Personenstatus” der Wert “AKTIV”.

## Request

Beim Erstellen eines Personenkontexts zu einem Personendatensatz wird immer die eigene Organisation per ID referenziert; die Organisation und deren ID {organisation.id} wird vom SchulConneX-Server aus dem access\_token übernommen.

Die folgende Tabelle listet die Attribute eines Personenkontext (“personendatensatz.personenkontexte”), welche von einem Quellsystem oder Dienst nicht verändert werden können.

Attribut	In den Anfrage Nutzdaten erforderlich?	Bemerkung
id	nein	ID des Personenkontexts wird vom SchulConneX-Server vergeben und ist eindeutig Dieses Attribut ist unveränderbar (immutable).
mandant	nein	ID des Mandanten, dem die Personen zugeordnet ist wird vom SchulConneX-Server vergeben und ist eindeutig Dieser Wert ist eine Referenz auf einen Mandanten.
organisation	nein	



organisation.id	nein	ID der Organisation  wird vom SchulConneX-Server vergeben und ist eindeutig  Dieses Attribut ist unveränderbar (immutable).
revision	nein	Revision des Personenkontexts  wird vom SchulConneX-Server mit der Erstellung des Datensatzes sowie Aktualisierung generiert  Dieser Wert kann nicht von Quellsystemen oder Diensten gesetzt werden.

*Tabelle 12: Liste von Attributen, welche durch ein Quellsystem oder Dienst über den Endpunkt `/v1/personen/{personendatensatz.person.id}/personenkontexte` auf HTTP-POST nicht gesetzt werden können.*

In Anbetracht der Attribute, die von einem Quellsystem oder Dienst bei der Erstellung eines Personenkontexts nicht gesetzt oder verändert werden können, ergibt sich folgende JSON-Struktur der Anfrage-Nutzdaten (Request Payload):

```
{
  "referrer": <String>, // ID des Personenkontexts im Quellsystem.
  "rolle": <Code>,
  // Siehe Code-Liste "Rolle". Dieses Feld ist ein Pflichtfeld.
  "personenstatus": <Code>, // Siehe Code-Liste "Personenstatus".
  "jahrgangsstufe": <Code> // Siehe Code-Liste "Jahrgangsstufe".
} // Siehe "Personenkontext".
```

*Quellcode 44: JSON-Struktur eines CREATE-Requests an `/v1/personen/{personendatensatz.person.id}/personenkontexte`*

Beispiel Anfrage-Nutzdaten (Request Payload):

```
{
  "referrer": "PerKt_12312",
  "rolle": "EXTERN",
  "personenstatus": "AKTIV"
}
```

*Quellcode 45: Beispiel Request zu `/personen/{personendatensatz.person.id}/personenkontexte` auf HTTP-POST*

## Response

Bei einer erfolgreichen Anforderung zum Erstellen eines Personenkontexts zu einem Personendatensatz wird diese Anforderung mit einer Repräsentation des Personenkontexts in den Antwort-Nutzdaten (Response Payload) und dem HTTP Status Code 200 („200 OK“) quittiert.

Nachfolgend die JSON-Struktur der Antwort-Nutzdaten (Response Payload):

```
{
  "id": <String>, // ID des Personenkontext.
  "referrer": <String>, // ID des Personenkontexts im Quellsystem.
  "mandant": <String>, // ID des Mandanten.
  "organisation": {
    "id": <String>, // ID der Organisation.
  }, // Siehe "Organisation".
  "rolle": <Code>, // Siehe Code-Liste "Rolle".
  "personenstatus": <Code>, // Siehe Code-Liste "Personenstatus".
  "jahrgangsstufe": <Code>, // Siehe Code-Liste "Jahrgangsstufe".
  "revision": <String> // Revision des Personenkontext.
} // Siehe "Personenkontext".
```

*Quellcode 46: JSON-Struktur einer CREATE-Response an /v1/personen/{personendatensatz.person.id}/personenkontexte*

Die Organisation wird beim Erstellen eines Personenkontexts vom SchulConneX-Server aus dem access\_token übernommen und per {personenkontext.organisation.id} referenziert. Die Antwort-Nutzdaten umfassen für das Attribut {personenkontext.organisation} mit dem Datentyp Organisation alle nicht optionalen Attribute – nur die ID {personenkontext.organisation.id}.

Beispiel Antwort-Nutzdaten (Response Payload):

```
{
  "id": "1d03016e-2798-43eb-8404-7e7d70424262",
  "referrer": "PerKt_12312",
  "mandant": "58f45270-8e54-40c6-a212-980307fc19be",
  "organisation": {
    "id": "c7de2c10-26a6-4d12-8152-f9e8db497625"
  },
  "rolle": "LERN",
  "personenstatus": "AKTIV",
  "revision": "1"
```

```
}
```

Quellcode 47: Beispiel Response zu `/personen/{personendatensatz.person.id}/personenkontexte` auf HTTP-POST

### Mögliche Fehlermeldungen

Zugrunde liegen die in 7.1.2 definierten Fehlermeldungen.

Darüber hinaus gibt es keine spezifischen Fehlermeldungen für diesen Endpunkt.

### READ

Dieser Schnittstellenendpunkt gibt die Personenkontexte zur angeforderten Person per ID `{personendatensatz.person.id}` zurück.

Ein READ kann nur mit einer HTTP-GET Anfrage erfolgen. Die Antwort-Nutzdaten (Response Payload) beinhalten ein Array von JSON-Objekten vom Datentyp Personenkontext, sofern Einträge (ggf. passend zu den optionalen Filtern) existieren. Das Datenmodell „Personenkontext“ ist in 5.4 beschrieben.

### Request

Die Anfrage erfordert keine Nutzdaten (Request Payload).

### Response

Nachfolgend die JSON-Struktur der Antwort-Nutzdaten (Response Payload):

```
[
  {
    "id": <String>, // ID des Personenkontext.
    "referrer": <String>, // ID des Personenkontexts im Quellsystem.
    "mandant": <String>, // ID des Mandanten.
    "organisation": {
      "id": <String>, // ID der Organisation.
    }, // Siehe "Organisation.
    "rolle": <Code>, // Siehe Code-Liste "Rolle".
    "personenstatus": <Code>,
    // Siehe Code-Liste "Personenstatus".
    "jahrgangsstufe": <Code> // Siehe Code-Liste "Jahrgangsstufe".
    "revision": <String> // Revision des Personenkontext.
  }
]
```

Quellcode 48: JSON-Struktur einer READ-Response an `/v1/personen/{personendatensatz.person.id}/personenkontexte`

Die Organisation wurde beim Erstellen eines Personenkontexts vom SchulConneX-Server aus dem access\_token übernommen und per {personenkontext.organisation.id} referenziert. Die Antwort-Nutzdaten umfassen standardmäßig für das Attribut {personenkontext.organisation} mit dem Datentyp Organisation alle nicht optionalen Attribute – nur die ID {personenkontext.organisation.id}.

Beispiel Antwort-Nutzdaten (Response Payload):

```
[{
  "id": "4d0f579c-0b9a-4d3a-b484-87b3bee8a2ad",
  "referrer": "PerKt_12312",
  "mandant": "58f45270-8e54-40c6-a212-980307fc19be",
  "organisation": {
    "id": "c7de2c10-26a6-4d12-8152-f9e8db497625"
  },
  "rolle": "LERN",
  "personenstatus": "AKTIV",
  "revision": "1"
},
{
  "id": "1d03016e-2798-43eb-8404-7e7d70424262",
  "referrer": "PerKt_12314",
  "mandant": "58f45270-8e54-40c6-a212-980307fc19be",
  "organisation": {
    "id": "c7de2c10-26a6-4d12-8152-f9e8db497625"
  },
  "rolle": "EXTERN",
  "personenstatus": "AKTIV",
  "revision": "1"
}
]
```

*Quellcode 49: Beispiel Response zu /personen/{personendatensatz.person.id}/personenkontexte auf HTTP-GET*

Durch das Setzen von Filterparametern (query-parameter) wird mit der HTTP-GET-Anfrage eine Filteroperation mitgeschickt, welche die Ergebnismenge der Antwort-Nutzdaten (Response Payload) beeinflusst.

Beispiel Antwort-Nutzdaten (Response Payload) mit den Filterparametern „?referrer=PerKt\_12314“:

```
[{
  "id": "1d03016e-2798-43eb-8404-7e7d70424262",
  "referrer": "PerKt_12314",
  "mandant": "58f45270-8e54-40c6-a212-980307fc19be",
  "organisation": {
    "id": "c7de2c10-26a6-4d12-8152-f9e8db497625"
  },
  "rolle": "LERN",
  "personenstatus": "AKTIV",
  "revision": "1"
}]
```

Quellcode 50: Beispiel Response zu `/personen/{personendatensatz.person.id}/personenkontexte` auf HTTP-GET mit den Filter-parametern (query-parameter) „?referrer=NI\_56-78-90\_123“

### Mögliche Fehlermeldungen

Zugrunde liegen die in 7.1.2 definierten Fehlermeldungen.

Darüber hinaus gibt es keine spezifischen Fehlermeldungen für diesen Endpunkt.

#### 10.1.4 `/personen/{id}/erreichbarkeiten` Erstellen und Lesen von zu einer Person gehörenden Erreichbarkeiten

Die Schnittstelle `/v1/personen/{.id}/erreichbarkeiten` erlaubt das Erstellen und Lesen von zu einer Person gehörenden Erreichbarkeiten.

Eine einzelne Erreichbarkeit besteht immer aus einem Objekt vom Datentyp 5.10 Kommunikation.

`https://subdomain.domain.org/subpaths/v1/personen/{id}/erreichbarkeiten`

Quellcode 51: URL-Struktur des Endpunkts `/v1/personen/{id}/erreichbarkeiten`

Der Pfad-Parameter `{id}` bezieht sich auf die ID der eigenen Person.

Bei Zugriffen auf den Endpunkt sind folgende Operationen und HTTP-Methoden erlaubt:

Operation	HTTP-Methode
CREATE	POST

READ	GET
------	-----

*Tabelle 13: Zugelassene Operationen auf dem Endpunkt /v1/personet/{id}/erreichbarkeiten*

## CREATE

Dieser Schnittstellenendpunkt erstellt eine Erreichbarkeit einer Person per ID {id} .

Ein CREATE ist nur mit HTTP-POST auf die API /v1/personen/{id}/erreichbarkeiten zum Erstellen von Erreichbarkeiten zu einer Person zugelassen. Die Anfrage-Nutzdaten (Request Payload) beinhalten ein JSON-Objekt des Datentyps Kommunikation. Das Datenmodell „Kommunikation“ ist in 5.10 beschrieben.

### Request

Die folgende Tabelle listet die Attribute einer Erreichbarkeit auf, welche von einem Quellsystem oder Dienst nicht verändert werden können.

Attribut	In den Anfrage Nutzdaten erforderlich?	Bemerkung
id	nein	ID der Erreichbarkeit wird vom SchulConneX-Server vergeben und ist eindeutig Dieses Attribut ist unveränderbar (immutable).
revision	nein	Revision der Erreichbarkeit wird vom SchulConneX-Server mit der Erstellung des Datensatzes sowie Aktualisierung generiert Dieser Wert kann nicht von Quellsystemen oder Diensten gesetzt werden.

*Tabelle 14: Liste von Attributen, welche durch ein Quellsystem oder Dienst über den Endpunkt /v1/personen/{id}/erreichbarkeiten auf HTTP-POST nicht gesetzt werden können.*

In Anbetracht der Attribute, die von einem Quellsystem oder Dienst bei der Erstellung einer Erreichbarkeit nicht gesetzt oder verändert werden können, ergibt sich folgende JSON-Struktur der Anfrage-Nutzdaten (Request Payload):

```
{
  "typ": <Code>, // Siehe Code-Liste Erreichbarkeitstyp
  "kennung": <String> // Kontaktadresse für den Erreichbarkeitstyp
}
```

*Quellcode 52: JSON-Struktur eines CREATE-Requests an /v1/personen/{id}/erreichbarkeiten*

### Beispiel Anfrage-Nutzdaten (Request Payload):

```
{
  "typ": "Email",
  "kennung": "sophie.maier@schiller-gymnasium.de"
}
```

*Quellcode 53: Beispiel Request zu /v1/personen/{id}/erreichbarkeiten auf HTTP-POST*

### Response

Bei einer erfolgreichen Anforderung zum Erstellen einer Erreichbarkeit einer Person wird diese Anforderung mit einer Repräsentation der Beziehung in den Antwort-Nutzdaten (Response Payload) und dem HTTP Status Code 200 („200 OK“) quittiert.

Nachfolgend die JSON-Struktur der Antwort-Nutzdaten (Response Payload):

```
{
  "id": <String>, // ID der Erreichbarkeit
  "typ": <code> // Siehe Code-Liste Erreichbarkeitstyp
  "kennung": <String> // Kontaktadresse für den Erreichbarkeitstyp
  "revision": <String>, // Revisionsnummer
}
```

*Quellcode 54: JSON-Struktur einer CREATE-Response an /v1/personen/{id}/erreichbarkeiten*

### Beispiel Antwort-Nutzdaten (Response Payload):

```
{
  "id": "7b33481a-7448-4f67-8722-77fff80850b6",
  "typ": "Email",
  "kennung": "sophie.maier@schiller-gymnasium.de"
  "revision": "1"
}
```

*Quellcode 55: Beispiel Response zu /v1/personen/{id}/erreichbarkeiten auf HTTP-POST*

### Mögliche Fehlermeldungen

Zugrunde liegen die in 7.1.2. definierten Fehlermeldungen, sowie spezifisch die Meldung 400/19 "Erreichbarkeit kann so nicht hinzugefügt werden."



## READ

Dieser Schnittstellenendpunkt gibt die Erreichbarkeiten zur angegebenen {id} zurück.

Ein READ ist nur mit HTTP-GET auf die API `/v1/personen/{id/erreichbarkeiten}` zum Abfragen der Erreichbarkeiten per {id} zugelassen. Die Antwort-Nutzdaten (Response Payload) beinhalten ein JSON-Objekt mit einem Array des Datentyps Kommunikation. Das Datenmodell „Kommunikation“ ist in 5.105.11 beschrieben.

Es werden nur Erreichbarkeiten zurück geliefert, welche direkt der Person zugeordnet sind. Sollten Personenkontexte dieser Person Erreichbarkeiten zugeordnet sein, so sind diese über die Schnittstelle `/personenkontexte/{personenkontext.id}/erreichbarkeiten` gesondert abzufragen.

### Request

Die Anfrage erfordert keine Nutzdaten (Request Payload).

### Response

Nachfolgend die JSON-Struktur der Antwort-Nutzdaten (Response Payload):

```
{ [
  {
    "id": <String>, // ID der Erreichbarkeit
    "typ": <code> // Siehe Code-Liste Erreichbarkeitstyp
    "kennung": <String> // Kontaktadresse für den Erreichbarkeitstyp
    "revision": <String>, // Revisionsnummer
  },
  ...
  {
    "id": <String>, // ID der Erreichbarkeit
    "typ": <code> // Siehe Code-Liste Erreichbarkeitstyp
    "kennung": <String> // Kontaktadresse für den Erreichbarkeitstyp
    "revision": <String>, // Revisionsnummer
  }
]}
```

*Quellcode 56: JSON-Struktur einer READ-Response an `/v1/personen/{id}/erreichbarkeiten`*

Beispiel Antwort-Nutzdaten (Response Payload):

```
{[
  {
    "id": "7b33481a-7448-4f67-8722-77fff80850b6",
    "typ": "Email",
    "kennung": "sophie.maier@schiller-gymnasium.de"
    "revision": "1"
  },
  {
    "id": "3ddedf03-9116-4d98-bcca-8a38e25419c4",
    "typ": "Email",
    "kennung": "sophie@maier.de"
    "revision": "1"
  }
]}
```

*Quellcode 57: Beispiel Response zu /personen/{id}/erreichbarkeiten*

### Mögliche Fehlermeldungen

Zugrunde liegen die in 7.1.2 definierten Fehlermeldungen.

Darüber hinaus gibt es keine spezifischen Fehlermeldungen für diesen Endpunkt.

### 10.1.5 [/personenkontexte] Personenkontexte der eigenen Organisation lesen

Die Schnittstelle /v1/personenkontexte bezieht sich auf das Anfordern und die Auflistung von Personenkontexten in Bezug zu der eigenen Organisation.

```
https://subdomain.domain.org/subpaths/v1/personenkontexte
```

Quellcode 58: URL-Struktur des Endpunkts personenkontexte

Bei Zugriffen auf den Endpunkt ist folgende Operation und HTTP-Methode erlaubt:

Operation	HTTP-Methode
READ	GET

Tabelle 15: Zugelassene Operation auf dem Endpunkt /v1/personenkontexte

Die Schnittstelle /v1/personenkontexte ermöglicht das Verwenden von Filterparametern zur Präzisierung der Anfrage bei HTTP-GET. Werden mehrere Filter angegeben, so sind sie als mit 'und' verknüpft anzusehen. Die mehrfache Nutzung des gleichen Filters in einer Anfrage ist nicht zulässig. Folgende Tabelle listet die zur Verfügung stehenden Filter auf:

Filter	Typ	Beschreibung
referrer	String	Möglichkeit, die Ergebnisliste der Anfrage nach dem Fremdschlüssel "referrer" zu filtern  Der SchulConneX-Server überprüft, ob der Wert des Filterparameters ohne Beachtung von Groß- und Kleinschreibung im Filter-Attribut beinhaltet ist (contains).
rolle	String (Code)	Möglichkeit, die Ergebnisliste der Anfrage nach dem Attribut "personenkontext.rolle" zu filtern  Der SchulConneX-Server überprüft, ob der Wert des Filterparameters mit dem Wert des Filter-Attributs übereinstimmt (equals).  Dem Filterparameter liegt eine Codeliste (Rolle) zugrunde.
personenstatus	String (Code)	Möglichkeit, die Ergebnisliste der Anfrage nach dem Attribut "personenkontext.personenstatus" zu filtern

Filter	Typ	Beschreibung
		<p>Der SchulConneX-Server überprüft, ob der Wert des Filterparameters mit dem Wert des Filter-Attributs übereinstimmt (equals).</p> <p>Dem Filterparameter liegt die Codeliste Personenstatus zugrunde.</p>
sichtfreigabe	String	<p>erlaubt es Personen, abhängig davon aufzulisten, ob diese von einer anderen Organisation zum Lesen freigegeben wurden</p> <p>Die möglichen Optionen sind "sichtfreigabe=ja", um nur die durch externe Organisationen freigegebenen Personen, beziehungsweise Personenkontexte aufzulisten, und "sichtfreigabe=nein", um nur eigene Personen beziehungsweise Personenkontexte aufzulisten. Wird nicht nach Sichtfreigabe gefiltert, so werden alle lesbaren Personen beziehungsweise Personenkontexte geliefert.</p>

*Tabelle 16: Zugelassene Filter auf dem Endpunkt /v1/personenkontexte*

## READ

Dieser Schnittstellenendpunkt gibt alle Personenkontexte zurück, auf die der Quellsystemanbieter zugreifen kann.

Ein READ kann nur mit einer HTTP-GET Anfrage erfolgen. Die Antwort-Nutzdaten (Response Payload) beinhalten ein Array von JSON-Objekten vom Datentyp Personendatensatz mit dem jeweiligen Personenkontext und der zugeordneten Person, sofern Einträge (ggf. passend zu den optionalen Filtern) existieren. Das Datenmodell „Personendatensatz“ ist in 5.5 beschrieben. Das Datenmodell „Person“ ist in 5.3 beschrieben. Das Datenmodell „Personenkontext“ ist in 5.4 beschrieben.

## Request

Die Anfrage erfordert keine Nutzdaten (Request Payload).

## Response

Nachfolgend die JSON-Struktur der Antwort-Nutzdaten (Response Payload):

```
[
  {
    "person": {
      "id": <String> // ID der Person.
    }, // Siehe "Person".
    "personenkontexte": [
      {
        "id": <String>, // ID des Personenkontext.
        "referrer": <String>,
          // ID des Personenkontext im Quellsystem.
        "mandant": <String>, // ID des Mandanten.
        "organisation": {
          "id": <String>, // ID der Organisation.
        }, // Siehe "Organisation".
        "rolle": <Code>, // Siehe Code-Liste "Rolle".
        "personenstatus": <Code>,
          // Siehe Code-Liste "Personenstatus".
        "jahrgangsstufe": <Code>,
          // Siehe Code-Liste "Jahrgangsstufe".
        "revision": <String>
          // Revision des Datensatzes Personenkontext.
      } // Siehe "Personenkontext".
    ]
  }
]
```

```
    } // Siehe "Personendatensatz".  
  ]
```

*Quellcode 59: JSON-Struktur einer READ-Response an /v1/personenkontexte*

Die Antwort-Nutzdaten umfassen standardmäßig für das Attribut {personendatensatz.person} mit dem Datentyp Person nur die ID über das Attribut {personenkontext.person.id}.

Die Antwort-Nutzdaten umfassen standardmäßig für das Attribut {personendatensatz.personenkontext.organisation} mit dem Datentyp Organisation alle nicht optionalen Attribute – also nur deren ID {personenkontext.organisation.id}.

Beispiel Antwort-Nutzdaten (Response Payload):

```
[{  
  "person": {  
    "id": "a6e1a860-8d44-4b2b-aef7-aa2c8bf5beb5"  
  },  
  "personenkontexte": [{  
    "id": "4d0f579c-0b9a-4d3a-b484-87b3bee8a2ad",  
    "referrer": "NI_12345_123",  
    "mandant": "58f45270-8e54-40c6-a212-980307fc19be",  
    "organisation": {  
      "id": "c7de2c10-26a6-4d12-8152-f9e8db497625"  
    },  
    "rolle": "LERN",  
    "personenstatus": "AKTIV",  
    "revision": "1"  
  }]  
},  
{  
  "person": {  
    "id": "a6e1a860-8d44-4b2b-aef7-aa2c8bf5beb5"  
  },  
  "personenkontexte": [{  
    "id": "1d03016e-2798-43eb-8404-7e7d70424262",  
    "referrer": "NI_56789_123",  
    "mandant": "58f45270-8e54-40c6-a212-980307fc19be",  
    "organisation": {  
      "id": "c7de2c10-26a6-4d12-8152-f9e8db497625"  
    },  
    "rolle": "LERN",  
    "personenstatus": "AKTIV",  
    "revision": "1"  
  }]  
}]
```

```

        "id": "c7de2c10-26a6-4d12-8152-f9e8db497625"
      },
      "rolle": "LERN",
      "personenstatus": "AKTIV",
      "revision": "1"
    ]
  }
]

```

*Quellcode 60: Beispiel Response zu /personenkontexte auf HTTP-GET*

Durch das Setzen von Filterparametern (query-parameter) wird mit der HTTP-GET-Anfrage eine Filteroperation mitgeschickt, welche die Ergebnismenge der Antwort-Nutzdaten (Response Payload) beeinflusst.

Beispiel Antwort-Nutzdaten (Response Payload) mit den Filterparametern „?referrer=NI\_56789\_123“:

```

[ {
  "person": {
    "id": "a6e1a860-8d44-4b2b-aef7-aa2c8bf5beb5"
  },
  "personenkontexte": [ {
    "id": "1d03016e-2798-43eb-8404-7e7d70424262",
    "referrer": "NI_56789_123",
    "mandant": "58f45270-8e54-40c6-a212-980307fc19be",
    "organisation": {
      "id": "c7de2c10-26a6-4d12-8152-f9e8db497625"
    },
    "rolle": "LERN",
    "personenstatus": "AKTIV",
    "revision": "1"
  } ]
} ]

```

*Quellcode 61: Beispiel Response zu personenkontexte auf HTTP-GET mit den Filter-parametern (query-parameter) „?referrer=NI\_56789\_123“*

### Mögliche Fehlermeldungen

Zugrunde liegen die in 7.1.2 definierten Fehlermeldungen.

Darüber hinaus gibt es keine spezifischen Fehlermeldungen für diesen Endpunkt.



### 10.1.6 [/personenkontexte/{id}] Individuelle Personenkontexte lesen, ändern und löschen

Die Schnittstelle /v1/personenkontexte/{personendatensatz.personenkontext.id} bezieht sich auf das Anfordern, Bearbeiten und Löschen eines Personenkontexts per ID.

```
https://subdomain.domain.org/subpaths/v1/personenkontexte/  
{personendatensatz.personenkontext.id}
```

Quellcode 62: URL-Struktur des Endpunkts personenkontexte/{personendatensatz.personenkontext.id}

Der Pfad-Parameter {personendatensatz.personenkontexte.id} bezieht sich auf die ID des Personenkontexts.

Bei Zugriffen auf den Endpunkt sind folgende Operationen und HTTP-Methoden erlaubt:

Operation	HTTP-Methode
READ	GET
UPDATE	PUT
DELETE	DELETE

Tabelle 17: Zugelassene Operationen auf dem Endpunkt /v1/personenkontexte/{personendatensatz.personenkontext.id}

#### READ

Dieser Schnittstellenendpunkt gibt den Personenkontext zur angegebenen {personendatensatz.personenkontexte.id} zurück.

Ein READ ist nur mit HTTP-GET auf die API /v1/personenkontexte/{personendatensatz.personenkontext.id} zum Abfragen eines Personenkontexts per ID zugelassen. Die Antwort-Nutzdaten (Response Payload) beinhalten ein Array von JSON-Objekten vom Datentyp Personendatensatz mit dem jeweiligen Personenkontext und der zugeordneten Person, sofern Einträge (ggf. passend zu den optionalen Filtern) existieren. Das Datenmodell „Personendatensatz“ ist in 5.5 beschrieben. Das Datenmodell „Person“ ist in 5.3 beschrieben. Das Datenmodell „Personenkontext“ ist in 5.4 beschrieben.

#### Request

Die Anfrage erfordert keine Nutzdaten (Request Payload).

## Response

Nachfolgend die JSON-Struktur der Antwort-Nutzdaten (Response Payload):

```
{
  "person": <Person>, // Siehe "Person".
  "personenkontexte": [
    <Personenkontext> // Siehe "Personenkontext".
  ]
}
```

Quellcode 63: JSON-Struktur der Antwort bei READ auf `/personendatensatz.person.id`

JSON-Struktur im Detail:

```
{
  "person": {
    <Person>, // Siehe "Person"
  },
  "personenkontexte": [
    {
      "id": <String>, // ID des Personenkontext.
      "referrer": <String>,
        // ID des Personenkontext im Quellsystem.
      "mandant": <String>, // ID des Mandanten.
      "organisation": {
        "id": <String>, // ID der Organisation.
      }, // Siehe "Organisation".
      "rolle": <Code>, // Siehe Code-Liste "Rolle".
      "personenstatus": <Code>,
        // Siehe Code-Liste "Personenstatus".
      "jahrgangsstufe": <Code>,
        // Siehe Code-Liste "Jahrgangsstufe".
      "revision": <String>
        // Revision des Datensatzes Personenkontext.
    } // Siehe "Personenkontext".
  ]
}
```

Quellcode 64: JSON-Struktur einer READ-Response an `/v1/personenkontexte/{id}`

Die Antwort-Nutzdaten umfassen standardmäßig für das Attribut {personendatensatz.personenkontext.organisation} mit dem Datentyp Organisation alle nicht optionalen Attribute – also nur deren ID {personendatensatz.personenkontext.organisation.id}.

Beispiel Antwort-Nutzdaten (Response Payload):

```
{
  "person": {
    "id": "a6e1a860-8d44-4b2b-aef7-aa2c8bf5beb5",
    "referrer": "125",
    "mandant": "58f45270-8e54-40c6-a212-980307fc19be",
    "name": {
      "familienname": "von Musterfrau",
      "vorname": "Natalie",
      "initialenfamilienname": "M",
      "initialenvorname": "N",
      "sortierindex": "4"
    },
    "geburt": {
      "datum": "2005-05-01",
      "geburtsort": "Berlin, Deutschland"
    },
    "geschlecht": "w",
    "lokalisierung": "de-DE",
    "vertrauensstufe": "VOLL",
    "revision": "1"
  },
  "personenkontexte": [{
    "id": "1d03016e-2798-43eb-8404-7e7d70424262",
    "referrer": "NI_56789_123",
    "mandant": "58f45270-8e54-40c6-a212-980307fc19be",
```

```
    "organisation": {  
        "id": "c7de2c10-26a6-4d12-8152-f9e8db497625"  
    },  
    "rolle": "LERN",  
    "personenstatus": "AKTIV",  
    "revision": "1"  
  }  
}
```

*Quellcode 65: Beispiel Response zu /personenkontexte/{personendatensatz.personenkontext.id} auf HTTP-GET*

### Mögliche Fehlermeldungen

Zugrunde liegen die in 7.1.2 definierten Fehlermeldungen.

Darüber hinaus gibt es keine spezifischen Fehlermeldungen für diesen Endpunkt.

## UPDATE

Dieser Schnittstellenendpunkt aktualisiert den Personenkontext zur angegebenen `{personendatensatz.personenkontexte.id}`.

Ein UPDATE ist nur mit HTTP-PUT auf die API `/v1/personenkontexte/{personendatensatz.personenkontext.id}` zum Aktualisieren eines Personenkontexts per ID zugelassen. Die Anfrage-Nutzdaten (Request Payload) beinhalten ein JSON-Objekt des Datentyps Personenkontext. Das Datenmodell „Personenkontext“ ist in 5.4 beschrieben.

Es ist erforderlich, dass zur Aktualisierungsanfrage im bearbeiteten Personenkontext das Attribut „personenkontext.revision“ des zugrunde liegenden Personenkontexts mitgeschickt wird. Der SchulConneX-Server überprüft anhand des mitgeschickten Wertes des Attributs „personenkontext.revision“, ob der Datensatz des Personenkontexts in der Zwischenzeit keine Änderung erfahren hat. Der Wert des Attributs kann von einem Quellsystem oder Dienst nicht verändert werden; dieser wird mit der Erstellung (Create) des Datensatzes Personenkontext und jeder erfolgreichen Änderung (Update) vom SchulConneX-Server neu generiert.

Die folgende Tabelle listet die Attribute eines Personenkontexts auf, welche von einem Quellsystem oder Dienst nicht verändert werden können.

Attribut	In den Anfrage Nutzdaten erforderlich?	Bemerkung
id	nein	ID des Personenkontexts wird vom SchulConneX-Server vergeben und ist eindeutig Dieses Attribut ist unveränderbar (immutable).
mandant	nein	ID des Mandanten, dem die Person zugeordnet ist wird vom SchulConneX-Server vergeben und ist eindeutig Dieser Wert ist eine Referenz auf einen Mandanten.
kommunikation	nein	wird über API <code>/erreichbarkeiten/{erreichbarkeit.id}</code> gesetzt oder entfernt

organisation	nein	
organisation.id	nein	ID der Organisation wird vom SchulConneX-Server vergeben und ist eindeutig Dieses Attribut ist unveränderbar (immutable).
rolle	nein	Rolle der Person innerhalb der Organisation Referenz auf einen Code der Codeliste (Rolle) Dieses Attribut ist unveränderbar (immutable).
sichtfreigabe	nein	gibt an, ob dieser Personenkontext nur aufgrund einer Freigabe durch eine andere Organisation sichtbar ist  Sollte der Wert von Sichtfreigabe "ja" sein, so führt ein Update auf diesen Personenkontext zu einer Fehlermeldung, da eine Sichtfreigabe nur das Lesen des Personenkontexts erlaubt.
revision	ja (Wert des für die Änderung zugrunde liegenden Personenkontext)	Revision des Personenkontexts wird vom SchulConneX-Server mit der Erstellung des Datensatzes sowie der Aktualisierung generiert Dieser Wert kann nicht von Quellsystemen oder Diensten gesetzt werden.

*Tabelle 18: Liste von Attributen, welche durch ein Quellsystem oder Dienst über den Endpunkt /v1/personenkontexte/{personendatensatz.personenkontext.id} auf HTTP-PUT nicht verändert werden können.*

Bis auf die Revisionsnummer können diese nicht veränderbaren Attribute im Request ausgelassen werden. Sind die Attribute im Request dennoch vorhanden, werden sie validiert und führen gegebenenfalls zu einer Fehlermeldung.

### Request

In Anbetracht der Attribute, die von einem Quellsystem oder Dienst bei der Aktualisierung eines Personenkontexts nicht gesetzt oder verändert werden können, ergibt sich folgende JSON-Struktur der Anfrage-Nutzdaten (Request Payload):

```
{
  "referrer": <String>, // ID im Quellsystem.
  "personenstatus": <Code>, // Siehe Code-Liste "Personenstatus".
  "jahrgangsstufe": <Code>, // Siehe Code-Liste "Jahrgangsstufe".
  "revision": <String>,
  // Revision des zugrunde liegenden Personenkontext.
  // Dieses Feld ist ein Pflichtfeld.
} // Siehe Personenkontext.
```

*Quellcode 66: JSON-Struktur eines UPDATE-Requests an /v1/personenkontexte/{id}*

#### Beispiel Anfrage-Nutzdaten (Request Payload):

```
{
  "referrer": "NI_56789_123",
  "personenstatus": "AKTIV",
  "revision": "1"
}
```

*Quellcode 67: Beispiel Request zu /personenkontexte/{personendatensatz.personenkontext.id} auf HTTP-PUT*

Anmerkung: Bei dieser Aktualisierungsanfrage handelt es sich um eine HTTP-PUT-Operation, bei der die Anfrage-Nutzlast mit den gegebenen Attributen der Aktualisierung zugrunde gelegt wird (mitgeschickte Attribute werden aktualisiert, nicht mitgeschickte Attribute werden gelöscht); es ist keine HTTP-PATCH-Operation, bei der lediglich die zu aktualisierenden Attribute in der Anfrage Nutzlast mitgeschickt werden.

#### Response

Bei einer erfolgreichen Anforderung zum Aktualisieren eines Personenkontexts wird diese Anforderung mit einer Repräsentation des Personenkontexts als Personendatensatz mit den beiden Attributen Person {personendatensatz.person} und Personenkontexte {personendatensatz.personenkontexte} in den Antwort-Nutzdaten (Response Payload) und dem HTTP Status Code 200 („200 OK“) quittiert.

Nachfolgend die JSON-Struktur der Antwort-Nutzdaten (Response Payload):

```
{
  "id": <String>, // ID des Personenkontext
  "referrer": <String>, // ID des Personenkontext im Quellsystem
  "mandant": <String>, // ID des Mandanten
  "organisation": {
```

```

        "id": <String>, // ID der Organisation
    }, // Siehe "Organisation"
    "rolle": <Code>, // Siehe Code-Liste "Rolle"
    "personenstatus": <Code>,
    // Siehe Code-Liste "Personenstatus"
    "jahrgangsstufe": <Code>,
    // Siehe Code-Liste "Jahrgangsstufe"
    "revision": <String>
    // Revision des Datensatzes Personenkontext
} // Siehe "Personenkontext"

```

Quellcode 68: JSON-Struktur einer UPDATE-Response an `/v1/personenkontexte/{id}`

Die Antwort-Nutzdaten umfassen standardmäßig für das Attribut `{personendatensatz.personenkontext.organisation}` mit dem Datentyp Organisation alle nicht optionalen Attribute – nur die ID `{personendatensatz.personenkontext.organisation.id}`.

Beispiel Antwort-Nutzdaten (Response Payload):

```

{
    "id": "1d03016e-2798-43eb-8404-7e7d70424262",
    "referrer": "NI_56789_123",
    "mandant": "58f45270-8e54-40c6-a212-980307fc19be",
    "organisation": {
        "id": "c7de2c10-26a6-4d12-8152-f9e8db497625"
    },
    "rolle": "LERN",
    "personenstatus": "AKTIV",
    "revision": "2"
}

```

Quellcode 69: Beispiel Response zu `/personenkontexte/{personendatensatz.personenkontext.id}` auf HTTP-PUT

### Mögliche Fehlermeldungen

Zugrunde liegen die in 7.1.2 definierten Fehlermeldungen.

Darüber hinaus gibt es keine spezifischen Fehlermeldungen für diesen Endpunkt.

### DELETE

Dieser Schnittstellenendpunkt löscht den Personenkontext zur angegebenen `{personendatensatz.personenkontexte.id}`.



Ein DELETE ist nur mit HTTP-DELETE auf die API `/v1/personenkontexte/{personendatensatz.personenkontext.id}` zum Löschen eines Personenkontexts per `{personendatensatz.personenkontext.id}` zugelassen. Die Anfrage-Nutzdaten (Request Payload) beinhalten ein JSON-Objekt des Datentyps Personenkontext. Das Datenmodell „Personenkontext“ ist in 5.4 beschrieben.

Es ist erforderlich, dass für eine Löschanfrage eines Personenkontexts das Attribut „personenkontext.revision“ des zugrunde liegenden Personenkontexts mitgeschickt wird. Der Anmeldeservice überprüft anhand des mitgeschickten Wertes des Attributs „personenkontext.revision“, ob der Datensatz des Personenkontexts in der Zwischenzeit keine Änderung erfahren hat.

### Request

Nachfolgend die JSON-Struktur der Anfrage-Nutzdaten (Request Payload):

```
{
    "revision": <String>,
    // Revision des zugrunde liegenden Personenkontext
} // Siehe Personenkontext
```

*Quellcode 55: JSON-Struktur eines DELETE-Requests an `/v1/personenkontexte/{id}`*

Beispiel Anfrage-Nutzdaten (Request Payload):

```
{
    "revision": "2"
}
```

*Quellcode 56: Beispiel Request zu `/personenkontexte/{personendatensatz.personenkontext.id}` auf HTTP-DELETE*

### Response

Bei einer erfolgreichen Ausführung der Löschanfrage wird es keine Antwort-Nutzdaten (Response Payload) geben. Eine erfolgreiche Löschanforderung eines Personenkontexts per ID wird mit dem HTTP Status Code 204 („204 No Content“) quittiert.

## Mögliche Fehlermeldungen

Zugrunde liegen die in 7.1.2 definierten Fehlermeldungen.

Ein Fehler, der nur bei dieser API auftreten kann, ist Error 400, Subcode 13: "Personenkontext wird genutzt." Dieser Fehler wird gemeldet, wenn Daten eines Personenkontexts bereits im Zusammenhang mit einem Anmeldevorgang an Dienste weitergeleitet wurden. In dem Fall muss die Löschung durch Setzen eines Löschezitpunkts erfolgen, um sicherzustellen, dass andere Dienste über die Löschung informiert werden und eventuell existierende Information bei den Diensten gelöscht werden können.

Darüber hinaus gibt es keine spezifischen Fehlermeldungen für diesen Endpunkt.

### 10.1.7 [/personenkontexte/{personenkontext.id}/beziehungen] Erstellen und Lesen von zu einem Personenkontext gehörenden Beziehungen

Die Schnittstelle /v1/personenkontexte/{personenkontext.id}/beziehungen erlaubt das Erstellen und Lesen von zu einem Personenkontext gehörenden Beziehungen.

Beziehungen sind immer von der Form "hat als". Besteht aus einem Personenkontext von Person A eine Beziehung vom Typ "SchB" zu einem Personenkontext (ktid) von Person B, so bedeutet dieses, dass Person A die Person B als Schulbegleiter (oder Schulbegleiterin) hat.

Im schulischen Gebrauch bedeutet das meistens, dass Beziehungen den Kontexten von Schülern und Schülerinnen zugeordnet sind. Schüler und Schülerinnen haben Betreuer oder Betreuerinnen, Tutoren oder Tutorinnen oder Erziehungsberechtigte. Der Personenkontext von beispielsweise Erziehungsberechtigten enthält keine Beziehungs-Relation.

```
https://subdomain.domain.org/subpaths/v1/personenkontexte/{personenkontext.id}/beziehungen
```

Quellcode 70: URL-Struktur des Endpunkts personenkontext/{personenkontext.id}/beziehungen

Der Pfad-Parameter {personenkontext.id} bezieht sich auf die ID des eigenen Personenkontextes.

Bei Zugriffen auf den Endpunkt sind folgende Operationen und HTTP-Methoden erlaubt:

Operation	HTTP-Methode
-----------	--------------

CREATE	POST
READ	GET

*Tabelle 19: Zugelassene Operationen auf dem Endpunkt /v1/personenkontext/{personenkontext.id}/beziehungen*

## CREATE

Dieser Schnittstellenendpunkt erstellt eine Beziehung aus einem Personenkontext per ID `{personenkontext.id}`.

Ein CREATE ist nur mit HTTP-POST auf die API `/v1/personenkontext/{personenkontext.id}/beziehungen` zum Erstellen von Beziehungen aus einem Personenkontext zugelassen. Die Anfrage-Nutzdaten (Request Payload) beinhalten ein JSON-Objekt des Datentyps Beziehung. Das Datenmodell „Beziehung“ ist in 5.11 beschrieben.

Bestehen zwischen zwei Personenkontexten mehrere Beziehungen (ist beispielsweise eine Person sowohl erziehungsberechtigt als auch betreuend für eine andere Person), so ist für jede Einzelbeziehung ein eigenes Beziehungsobjekt zu erstellen.

## Request

Die folgende Tabelle listet die Attribute einer Beziehung auf, welche von einem Quellsystem oder Dienst nicht verändert werden können.

Attribut	In den Anfrage Nutzdaten erforderlich?	Bemerkung
id	nein	ID der Beziehung wird vom SchulConneX-Server vergeben und ist eindeutig Dieses Attribut ist unveränderbar (immutable).
revision	nein	Revision der Beziehung wird vom SchulConneX-Server mit der Erstellung des Datensatzes sowie Aktualisierung generiert Dieser Wert kann nicht von Quellsystemen oder Diensten gesetzt werden.

*Tabelle 20: Liste von Attributen, welche durch ein Quellsystem oder Dienst über den Endpunkt `/v1/personenkontexte/{personenkontext.id}/bezugscontext` auf HTTP-POST nicht gesetzt werden können.*

In Anbetracht der Attribute, die von einem Quellsystem oder Dienst bei der Erstellung einer Beziehung nicht gesetzt oder verändert werden können, ergibt sich folgende JSON-Struktur der Anfrage-Nutzdaten (Request Payload):

```
{
    "ktid": <String>, // ID des Personenkontexts zu dem die Beziehung
    besteht
    "beziehung": <Code> // Siehe Code-Liste Beziehungen
}
```

*Quellcode 71: JSON-Struktur eines CREATE-Requests an /v1/personenkontexte/{personenkontext.id}/beziehungen*

#### Beispiel Anfrage-Nutzdaten (Request Payload):

```
{
    "ktid": " e3bbcd03-7d91-4474-9de3-28a6bd162005",
    "beziehung": "SorgBer"
}
```

*Quellcode 72: Beispiel Request zu /v1/personenkontexte/{personenkontext.id }/beziehungen auf HTTP-POST*

#### Response

Bei einer erfolgreichen Anforderung zum Erstellen einer Beziehung zu einem Personenkontext wird diese Anforderung mit einer Repräsentation der Beziehung in den Antwort-Nutzdaten (Response Payload) und dem HTTP Status Code 200 („200 OK“) quittiert.

Nachfolgend die JSON-Struktur der Antwort-Nutzdaten (Response Payload):

```
{
    "id": <String>, // ID der Beziehung
    "ktid": <String>, // ID des Personenkontexts zu dem die Beziehung
    besteht
    "beziehung": <Code>, // Siehe Code-Liste Beziehungen
    "revision": <String>, // Revisionsnummer
}
```

*Quellcode 73: JSON-Struktur einer CREATE-Response an /v1/personenkontexte/{personenkontext.id}/beziehungen*

#### Beispiel Antwort-Nutzdaten (Response Payload):

```
{
    "id": "d7711bd0-65d6-441a-b486-9f0e95847156",
    "ktid": "46350f47-9739-4708-9a3c-b57bd4c0fc2b ",
    "beziehung": "SorgBer",
}
```

```
"revision": "1"
}
```

Quellcode 74: Beispiel Response zu `/v1/personenkontexte/{personenkontext.id}/beziehungen` auf HTTP-POST

### Mögliche Fehlermeldungen

Zugrunde liegen die in 7.1.2. definierten Fehlermeldungen.

Ein Fehler, der nur bei dieser API auftreten kann, ist Error 400, Subcode 18: ""Beziehung darf so nicht erstellt werden." Dieser Fehler wird gemeldet, wenn die zu erstellende Beziehung nicht unterstützt werden kann, beispielsweise, da Minderjährige nicht sorgeberechtigt sein können.

### READ

Dieser Schnittstellenendpunkt gibt die Beziehungen zur angegebenen `{personenkontext.id}` zurück.

Ein READ ist nur mit HTTP-GET auf die API `/v1/personenkontexte/{personenkontext.id}/beziehungen` zum Abfragen der Beziehung per `personenkontext.id` zugelassen. Die Antwort-Nutzdaten (Response Payload) beinhalten ein JSON-Objekt mit einem Array des Datentyps Beziehung. Das Datenmodell „Beziehung“ ist in 5.11 beschrieben.

Die Schnittstelle `/v1/personenkontexte/{personenkontext.id}/beziehungen` ermöglicht das Verwenden von Filterparametern zur Präzisierung der Anfrage bei HTTP-GET.

Folgende Tabelle listet die zur Verfügung stehenden Filter auf:

Filter	Typ	Beschreibung
hat_als_beziehung	String (Boolean)	listet alle Beziehungen auf, welche der, durch die Personenkontext-ID gegebenen, Personenkontext hat  Die Beziehungen werden unter 'hat_als_beziehungen' aufgelistet.  Der Defaultwert ist "ja", es werden also ohne Filterangabe alle Beziehungen angegeben, welche ein Personenkontext hat.
ist_von_beziehungen	String (Boolean)	listet alle Beziehungen auf, welche von anderen Personenkontexten zu dem, durch die Personenkontext-ID gegebenen, Personenkontext bestehen. Die

		<p>Beziehungen werden unter 'ist_von_beziehungen' aufgelistet.</p> <p>Der Defaultwert ist "nein", es werden also ohne Filterangabe keine Beziehungen angegeben, welche zu dem gegebenen Personenkontext bestehen.</p>
--	--	---

*Tabelle 21: Zugelassene Filter auf dem Endpunkt /v1/personenkontexte/{personenkontext.id}/beziehungen*

## READ

Dieser Schnittstellenendpunkt gibt die Beziehungen zurück, die vom Anfrager ausgelesen werden dürfen.

### Request

Die Anfrage erfordert keine Nutzdaten (Request Payload).

### Response

Nachfolgend die JSON-Struktur der Antwort-Nutzdaten (Response Payload):

```
{
  "hat_als_beziehungen": [ // Array von Beziehungen, welche von dem
                           Personenkontext ausgehen
    {
      "id": <String>, // ID der Beziehung.
      "ktid": <Personenkontext-ID welche der Bezug in
              der angegebenen Rolle ist>
      "beziehung": <Art der Beziehung>
    },
    ...
  ],
  "ist_von_beziehungen": [
    {
      "id": <String>, // ID der Beziehung.
      "ktid": <Personenkontext-ID welche der Bezug in
              der angegebenen Rolle ist>
      "beziehung": <Art der Beziehung>
    }
  ],
}
```

*Quellcode 75: JSON-Struktur einer READ-Response an /v1/ personenkontexte/{personenkontext.id}/beziehungen*

Beispiel Antwort-Nutzdaten (Response Payload):

```
{
  "hat_als_beziehungen": [
    {
```



```

        "id": "70d413eb-4bc0-424d-b99a-f1bef3b8968f",
        "ktid": "faf0ffa2-65ea-466c-b198-52f3a9210358",
        "beziehung": "SchB"
    },
    {
        "id": "70d413eb-4bc0-424d-b99a-f1bef3b8968f",
        "ktid": "68d1d77c-c808-4c37-b5b3-e5a551bb1930",
        "beziehung": "SchB"
    }
],
}

```

Quellcode 76: Beispiel Response zu `/personenkontext/{personenkontext.id}/beziehungen` auf HTTP-GET ohne Filter

### Beispiel Antwort-Nutzdaten (Response Payload):

```

{
  "hat_als_beziehungen": [
    {
      "id": "70d413eb-4bc0-424d-b99a-f1bef3b8968f",
      "ktid": "faf0ffa2-65ea-466c-b198-52f3a9210358",
      "beziehung": "SchB"
    },
    {
      "id": "b5342223-0a15-4e48-b1a5-1b8876093c42",
      "ktid": "68d1d77c-c808-4c37-b5b3-e5a551bb1930",
      "beziehung": "SchB"
    }
  ],
  "ist_von_beziehungen": [
    {
      "id": "b19d3cf5-b493-4d50-b03c-3e6e4f0c6bbe",
      "ktid": "aafad4cd-b1e6-425d-888e-8785ade82e05",
      "beziehung": "SorgBer"
    }
  ],
}

```

Quellcode 77: Beispiel Response zu `/personenkontext/{personenkontext.id}/beziehungen` auf HTTP-GET mit Filter `"?hat_als_beziehungen=ja&ist_von_beziehungen=ja"`

## Mögliche Fehlermeldungen

Zugrunde liegen die in 7.1.2 definierten Fehlermeldungen.

Darüber hinaus gibt es keine spezifischen Fehlermeldungen für diesen Endpunkt.

### 10.1.8 [/beziehungen/{beziehung.id}] Lesen und löschen einer einzelnen Beziehung

Die Schnittstelle /v1/beziehungen/{beziehung.id} bezieht sich auf das Anfordern einer Beziehung per ID.

```
https://subdomain.domain.org/subpaths/v1/beziehungen/{beziehung.id}
```

Quellcode 78: URL-Struktur des Endpunkts /v1/beziehungen/{beziehung.id}

Der Pfad-Parameter {beziehung.id} bezieht sich auf die ID der Beziehung.

Bei Zugriffen auf den Endpunkt sind folgende Operationen und HTTP-Methoden erlaubt:

Operation	HTTP-Methode
READ	GET
DELETE	DELETE

Tabelle 22: Zugelassene Operationen auf dem Endpunkt /v1/beziehungen/{beziehung.id}

Beziehungen können nicht direkt geändert werden. Sollte sich die Beziehung ändern, wenn beispielsweise eine andere Person das Sorgerecht bekommt oder sich die Rolle in einer Beziehung ändert, so ist die alte Beziehung zu löschen und eine neue Beziehung zu erstellen.

#### READ

Dieser Schnittstellenendpunkt gibt die Beziehung zur angegebenen {beziehung.id} zurück.

Ein READ ist nur mit HTTP-GET auf die API /v1/beziehungen/{beziehung.id} zum Abfragen einer Beziehung per ID zugelassen. Die Antwort-Nutzdaten (Response Payload) beinhalten ein JSON-Objekt des Datentyps Beziehung. Das Datenmodell „Beziehung“ ist in 5.11 beschrieben.

Zusätzlich zu den Informationen aus dem Datenmodell wird ein Attribut `ist_von_ktid` mitgeliefert, welches angibt, welchen Personenkontext die Beziehung referenziert.

#### Request

Die Anfrage erfordert keine Nutzdaten (Request Payload).

## Response

Nachfolgend die JSON-Struktur der Antwort-Nutzdaten (Response Payload):

```
{
  "id": <String>, // ID der Beziehung
  "ist_von_ktid": <String>, // ID des Personenkontexts aus dem heraus die
  Beziehung besteht
  "ktid": <String>, // ID des Personenkontexts zu dem die Beziehung
  besteht
  "beziehung": <Code>, // Siehe Code-Liste Beziehungen
  "revision": <String> // Revisionsnummer
}
```

*Quellcode 79: JSON-Struktur einer READ-Response an /v1/beziehungen/{beziehung.id}*

Beispiel Antwort-Nutzdaten (Response Payload):

```
{
  "id": "70e9ae0d-80e3-4169-9698-f97fc03edcf8",
  "ist_von_ktid": "846b1a6e-4079-4a73-a434-0f1e4751889b",
  "ktid": "70e9ae0d-80e3-4169-9698-f97fc03edcf8",
  "beziehung": "SorgBer",
  "revision": "1"
}
```

*Quellcode 80: Beispiel Response zu /v1/beziehungen/{beziehung.id} auf HTTP-GET*

## Mögliche Fehlermeldungen

Zugrunde liegen die in 7.1.2. definierten Fehlermeldungen.

Darüber hinaus gibt es keine spezifischen Fehlermeldungen für diesen Endpunkt.

## DELETE

Dieser Schnittstellenendpunkt löscht die Beziehung zur angegebenen {beziehung.id}.

Ein DELETE ist nur mit HTTP-DELETE auf die API /v1/beziehungen/{beziehung.id} zum Löschen einer Beziehung per {beziehung.id} zugelassen.

Es ist erforderlich, dass für eine Löschanfrage einer Beziehung das Attribut "revision" der zugrunde liegenden Beziehung mitgeschickt wird. Der SchulConneX-Server überprüft anhand des mitgeschickten Wertes des Attributs "revision", ob der Datensatz in der Zwischenzeit keine Änderung erfahren hat.

## Request

Nachfolgend die JSON-Struktur der Anfrage-Nutzdaten (Request Payload):

```
{
  "revision": <String>,
  // Revision der zugrunde liegenden Beziehung
}
```

*Quellcode 81: JSON-Struktur eines DELETE-Requests an /v1/beziehungen/{beziehung.id}*

Beispiel Anfrage-Nutzdaten (Request Payload):

```
{
  "revision": "2"
}
```

*Quellcode 82: Beispiel Request zu /v1/beziehungen/{beziehung.id} auf HTTP-DELETE*

## Response

Bei einer erfolgreichen Ausführung der Löschanfrage wird es keine Antwort-Nutzdaten (Response Payload) geben. Eine erfolgreiche Löschanforderung einer Beziehung per ID wird mit dem HTTP Status Code 204 („204 No Content“) quittiert.

## Mögliche Fehlermeldungen

Zugrunde liegen die in 7.1.2 definierten Fehlermeldungen.

Darüber hinaus gibt es keine spezifischen Fehlermeldungen für diesen Endpunkt.

Fehlermeldungen für diesen Endpunkt.

### 10.1.9 [/personenkontexte/{personenkontext.id}/erreichbarkeiten] Erstellen und Lesen von zu einem Personenkontext gehörenden Erreichbarkeiten

Die Schnittstelle /v1/personenkontexte/{personenkontext.id}/erreichbarkeiten erlaubt das Erstellen und Lesen von zu einem Personenkontext gehörenden Erreichbarkeiten.

Eine einzelne Erreichbarkeit besteht immer aus einem Objekt vom Datentyp 5.10 Kommunikation.

```
https://subdomain.domain.org/subpaths/v1/personenkontexte/{personenkontext.id}/erreichbarkeiten
```

Quellcode 83: URL-Struktur des Endpunkts `personenkontext/{personenkontext.id}/erreichbarkeiten`

Der Pfad-Parameter `{personenkontext.id}` bezieht sich auf die ID des eigenen Personenkontextes.

Bei Zugriffen auf den Endpunkt sind folgende Operationen und HTTP-Methoden erlaubt:

Operation	HTTP-Methode
CREATE	POST
READ	GET

Tabelle 23: Zugelassene Operationen auf dem Endpunkt `/v1/personenkontext/{personenkontext.id}/erreichbarkeiten`

## CREATE

Dieser Schnittstellenendpunkt erstellt eine Erreichbarkeit aus einem Personenkontext per ID `{personenkontext.id}`.

Ein CREATE ist nur mit HTTP-POST auf die API `/v1/personenkontext/{personenkontext.id}/erreichbarkeiten` zum Erstellen von Erreichbarkeiten aus einem Personenkontext zugelassen. Die Anfrage-Nutzdaten (Request Payload) beinhalten ein JSON-Objekt des Datentyps Kommunikation. Das Datenmodell „Kommunikation“ ist in 5.11 beschrieben.

### Request

Die folgende Tabelle listet die Attribute einer Erreichbarkeit auf, welche von einem Quellsystem oder Dienst nicht verändert werden können.

Attribut	In den Anfrage Nutzdaten erforderlich?	Bemerkung
id	nein	ID der Erreichbarkeit wird vom SchulConneX-Server vergeben und ist eindeutig Dieses Attribut ist unveränderbar (immutable).
revision	nein	Revision der Erreichbarkeit wird vom SchulConneX-Server mit der Erstellung des Datensatzes sowie Aktualisierung generiert Dieser Wert kann nicht von Quellsystemen oder Diensten gesetzt werden.

*Tabelle 24: Liste von Attributen, welche durch ein Quellsystem oder Dienst über den Endpunkt `/v1/personenkontexte/{personenkontext.id}/erreichbarkeiten` auf HTTP-POST nicht gesetzt werden können.*

In Anbetracht der Attribute, die von einem Quellsystem oder Dienst bei der Erstellung einer Erreichbarkeit nicht gesetzt oder verändert werden können, ergibt sich folgende JSON-Struktur der Anfrage-Nutzdaten (Request Payload):

```
{
  "typ": <Code>, // Siehe Code-Liste Erreichbarkeitstyp
  "kennung": <String> // Kontaktadresse für den Erreichbarkeitstyp
```

```
}
```

*Quellcode 84: JSON-Struktur eines CREATE-Requests an /v1/personenkontexte/{personenkontext.id}/erreichbarkeiten*

#### Beispiel Anfrage-Nutzdaten (Request Payload):

```
{
  "typ": "Email",
  "kennung": "sophie.maier@schiller-gymnasium.de"
}
```

*Quellcode 85: Beispiel Request zu /v1/personenkontexte/{personenkontext.id}/erreichbarkeiten auf HTTP-POST*

#### Response

Bei einer erfolgreichen Anforderung zum Erstellen einer Erreichbarkeit zu einem Personenkontext wird diese Anforderung mit einer Repräsentation der Beziehung in den Antwort-Nutzdaten (Response Payload) und dem HTTP Status Code 200 („200 OK“) quittiert.

Nachfolgend die JSON-Struktur der Antwort-Nutzdaten (Response Payload):

```
{
  "id": <String>, // ID der Erreichbarkeit
  "typ": <code> // Siehe Code-Liste Erreichbarkeitstyp
  "kennung": <String> // Kontaktadresse für den Erreichbarkeitstyp
  "revision": <String>, // Revisionsnummer
}
```

*Quellcode 86: JSON-Struktur einer CREATE-Response an /v1/personenkontexte/{personenkontext.id}/erreichbarkeiten*

#### Beispiel Antwort-Nutzdaten (Response Payload):

```
{
  "id": "7b33481a-7448-4f67-8722-77fff80850b6",
  "typ": "Email",
  "kennung": "sophie.maier@schiller-gymnasium.de"
  "revision": "1"
}
```

*Quellcode 87: Beispiel Response zu /v1/personenkontexte/{personenkontext.id}/erreichbarkeiten auf HTTP-POST*

## Mögliche Fehlermeldungen

Zugrunde liegen die in 7.1.2. definierten Fehlermeldungen, sowie spezifisch die Meldung 400/19 "Erreichbarkeit kann so nicht hinzugefügt werden."

## READ

Dieser Schnittstellenendpunkt gibt die Erreichbarkeiten zur angegebenen `{personenkontext.id}` zurück.

Ein READ ist nur mit HTTP-GET auf die API `/v1/personenkontexte/{personenkontext.id}/erreichbarkeiten` zum Abfragen der Erreichbarkeiten per `personenkontext.id` zugelassen. Die Antwort-Nutzdaten (Response Payload) beinhalten ein JSON-Objekt mit einem Array des Datentyps Kommunikation. Das Datenmodell „Kommunikation“ ist in 5.10 beschrieben.

## Request

Die Anfrage erfordert keine Nutzdaten (Request Payload).

## Response

Nachfolgend die JSON-Struktur der Antwort-Nutzdaten (Response Payload):

```
{ [
  {
    "id": <String>, // ID der Erreichbarkeit
    "typ": <code> // Siehe Code-Liste Erreichbarkeitstyp
    "kennung": <String> // Kontaktadresse für den Erreichbarkeitstyp
    "revision": <String>, // Revisionsnummer
  },
  ...
  {
    "id": <String>, // ID der Erreichbarkeit
    "typ": <code> // Siehe Code-Liste Erreichbarkeitstyp
    "kennung": <String> // Kontaktadresse für den Erreichbarkeitstyp
    "revision": <String>, // Revisionsnummer
  }
]}
```

*Quellcode 88: JSON-Struktur einer READ-Response an `/v1/personenkontexte/{personenkontext.id}/erreichbarkeiten`*



### Beispiel Antwort-Nutzdaten (Response Payload):

```
{[
  {
    "id": "7b33481a-7448-4f67-8722-77fff80850b6",
    "typ": "Email",
    "kennung": "sophie.maier@schiller-gymnasium.de"
    "revision": "1"
  },
  {
    "id": "3ddedf03-9116-4d98-bcca-8a38e25419c4",
    "typ": "Email",
    "kennung": "sophie@maier.de"
    "revision": "1"
  }
]}
```

Quellcode 89: Beispiel Response zu `/personenkontext/{personenkontext.id}/erreichbarkeiten`

### Mögliche Fehlermeldungen

Zugrunde liegen die in 7.1.2 definierten Fehlermeldungen.

Darüber hinaus gibt es keine spezifischen Fehlermeldungen für diesen Endpunkt.

#### 10.1.10 `/erreichbarkeiten/{erreichbarkeit.id}` Lesen und löschen einer einzelnen Erreichbarkeit

Die Schnittstelle `/v1/erreichbarkeiten/{erreichbarkeit.id}` bezieht sich auf das Anfordern einer Erreichbarkeit per ID.

```
https://subdomain.domain.org/subpaths/v1/erreichbarkeiten/{erreichbarkeit.id}
```

Quellcode 90: URL-Struktur des Endpunkts `/v1/erreichbarkeiten/{erreichbarkeit.id}`

Der Pfad-Parameter `{erreichbarkeit.id}` bezieht sich auf die ID der Erreichbarkeit.

Bei Zugriffen auf den Endpunkt sind folgende Operationen und HTTP-Methoden erlaubt:

Operation	HTTP-Methode
READ	GET

DELETE	DELETE
--------	--------

Tabelle 25: Zugelassene Operationen auf dem Endpunkt `/v1/erreichbarkeiten/{erreichbarkeit.id}`

Erreichbarkeiten können nicht direkt geändert werden. Sollte sich eine Erreichbarkeit ändern, wenn beispielsweise eine Person die E-Mail-Adresse wechselt, so ist die alte Erreichbarkeit zu löschen und eine neue Erreichbarkeit zu erstellen.

### READ

Dieser Schnittstellenendpunkt gibt die Erreichbarkeit zur angegebenen `{erreichbarkeit.id}` zurück.

Ein READ ist nur mit HTTP-GET auf die API `/v1/erreichbarkeit/{erreichbarkeit.id}` zum Abfragen einer Erreichbarkeit per ID zugelassen. Die Antwort-Nutzdaten (Response Payload) beinhalten ein JSON-Objekt des Datentyps Kommunikation. Das Datenmodell „Kommunikation“ ist in 5.105.11 beschrieben.

### Request

Die Anfrage erfordert keine Nutzdaten (Request Payload).

### Response

Nachfolgend die JSON-Struktur der Antwort-Nutzdaten (Response Payload):

```
{
  "id": <String>, // ID der Erreichbarkeit
  "typ": <code> // Siehe Code-Liste Erreichbarkeitstyp
  "kennung": <String> // Kontaktadresse für den Erreichbarkeitstyp
  "revision": <String>, // Revisionsnummer
}
```

Quellcode 91: JSON-Struktur einer READ-Response an `/v1/erreichbarkeiten/{erreichbarkeitbeziehung.id}`

Beispiel Antwort-Nutzdaten (Response Payload):

```
{
  "id": "3ddedf03-9116-4d98-bcca-8a38e25419c4",
  "typ": "Email",
  "kennung": "sophie@maier.de"
  "revision": "1"
```

```
}
```

Quellcode 92: Beispiel Response zu `/v1/erreichbarkeiten/{erreichbarkeit.id}` auf HTTP-GET

### Mögliche Fehlermeldungen

Zugrunde liegen die in 7.1.2. definierten Fehlermeldungen.

Darüber hinaus gibt es keine spezifischen Fehlermeldungen für diesen Endpunkt.

### DELETE

Dieser Schnittstellenendpunkt löscht die Erreichbarkeit zur angegebenen `{erreichbarkeit.id}`.

Ein DELETE ist nur mit HTTP-DELETE auf die API `/v1/erreichbarkeiten/{erreichbarkeit.id}` zum Löschen einer Erreichbarkeit per `{erreichbarkeit.id}` zugelassen.

Es ist erforderlich, dass für eine Löschanfrage einer Beziehung das Attribut "revision" der zugrunde liegenden Erreichbarkeit mitgeschickt wird. Der SchulConneX-Server überprüft anhand des mitgeschickten Wertes des Attributs "revision", ob der Datensatz in der Zwischenzeit keine Änderung erfahren hat.

### Request

Nachfolgend die JSON-Struktur der Anfrage-Nutzdaten (Request Payload):

```
{
  "revision": <String>,
  // Revision der zugrunde liegenden Erreichbarkeit
}
```

Quellcode 93: JSON-Struktur eines DELETE-Requests an `/v1/erreichbarkeiten/{erreichbarkeit.id}`

Beispiel Anfrage-Nutzdaten (Request Payload):

```
{
  "revision": "2"
}
```

Quellcode 94: Beispiel Request zu `/v1/erreichbarkeiten/{erreichbarkeit.id}` auf HTTP-DELETE

### Response

Bei einer erfolgreichen Ausführung der Löschanfrage wird es keine Antwort-Nutzdaten (Response Payload) geben. Eine erfolgreiche Löschanforderung einer Erreichbarkeit per ID wird mit dem HTTP Status Code 204 („204 No Content“) quittiert.

## Mögliche Fehlermeldungen

Zugrunde liegen die in 7.1.2 definierten Fehlermeldungen.

Darüber hinaus gibt es keine spezifischen Fehlermeldungen für diesen Endpunkt.

## 10.2 Schnittstellen für Organisationen

### 10.2.1 [/organisationen] Organisationen auflisten

Die Schnittstelle /v1/organisationen bezieht sich auf die Auflistung von Organisationen.

<code>https://subdomain.domain.org/subpaths/v1/organisationen</code>
--

Quellcode 95: URL-Struktur des Endpunkts organisationen

Bei Zugriffen auf den Endpunkt ist folgende Operation und HTTP-Methode erlaubt:

Operation	HTTP-Methode
READ	GET

Tabelle 26: Zugelassene Operation auf dem Endpunkt /v1/organisationen

Die Schnittstelle /v1/organisationen ermöglicht das Verwenden von Filterparametern zur Präzisierung der Anfrage bei HTTP-GET. Werden mehrere Filter angegeben, so sind sie als mit 'und' verknüpft anzusehen. Die mehrfache Nutzung des gleichen Filters in einer Anfrage ist nicht zulässig. Folgende Tabelle listet die zur Verfügung stehenden Filter auf:

Filter	Typ	Beschreibung
kennung	String	Möglichkeit, die Ergebnisliste der Anfrage nach "kennung" zu filtern  Der SchulConneX-Server überprüft, ob der Wert des Filterparameters ohne Beachtung von Groß- und Kleinschreibung im Filter-Attribut beinhaltet ist (contains).
name	String	Möglichkeit, die Ergebnisliste der Anfrage nach dem Attribut "organisation.name" zu filtern

		Der SchulConneX-Server überprüft, ob der Wert des Filterparameters ohne Beachtung von Groß- und Kleinschreibung im Filter-Attribut beinhaltet ist (contains).
typ	String (Code)	<p>Möglichkeit, die Ergebnisliste der Anfrage nach dem Attribut "organisation.typ" zu filtern</p> <p>Der SchulConneX-Server überprüft, ob der Wert des Filterparameters mit dem Wert des Filter-Attributs übereinstimmt (equals).</p> <p>Dem Filterparameter liegt eine Codeliste (Organisationstyp) zugrunde.</p>

*Tabelle 27: Zugelassene Filter auf dem Endpunkt /v1/organisationen*

## READ

Dieser Schnittstellenendpunkt gibt die Organisationen zurück, die vom Anfrager ausgelesen werden dürfen.

Ein READ kann nur mit einer HTTP-GET Anfrage erfolgen. Die Antwort-Nutzdaten (Response Payload) beinhalten ein Array von JSON-Objekten vom Datentyp Organisation, sofern Einträge (ggf. passend zu den optionalen Filtern) existieren. Das Datenmodell „Organisation“ ist in 5.2 beschrieben.

### Request

Die Anfrage erfordert keine Nutzdaten (Request Payload).

### Response

Nachfolgend die JSON-Struktur der Antwort-Nutzdaten (Response Payload):

```
[  
  <Organisation> // Siehe "Organisation"  
]
```

*Quellcode 96: JSON-Struktur einer READ-Response an /v1/organisationen*

Beispiel Antwort Nutzdaten (Response Payload):

```
[ {  
  "id": "c7de2c10-26a6-4d12-8152-f9e8db497625",  
  "kennung": "NI_12345",  
  "name": "Muster-Schule",  
  "typ": "SCHULE",  
  "traegerschaft": "02"  
 },  
 {  
  "id": "d4de2e70-41b2-4a55-9325-a7e4db497934",  
  "kennung": "RP_A123456",  
  "name": "Beispiel-Schule",  
  "typ": "SCHULE",  
  "traegerschaft": "02"  
 }  
 ]
```

*Quellcode 97: Beispiel Response zu /organisationen auf HTTP-GET*

Durch das Setzen von Filterparametern (query-parameter) wird mit der HTTP-GET-Anfrage eine Filteroperation mitgeschickt, welche die Ergebnismenge der Antwort-Nutzdaten (Response Payload) beeinflusst.

Beispiel Antwort-Nutzdaten (Response Payload) mit den Filterparametern „?kennung=NI\_12345&typ=SCHULE“:

```
[ {  
  "id": "c7de2c10-26a6-4d12-8152-f9e8db497625",  
  "kennung": "NI_12346",  
  "name": "Muster-Schule",  
  "typ": "SCHULE",  
  "traegerschaft": "02"  
}]
```

*Quellcode 98: Beispiel Response zu /organisationen auf HTTP-GET mit den Filter-parametern „?kennung=NI\_12-34-56&typ=SCHULE“*

### Mögliche Fehlermeldungen

Zugrunde liegen die in 7.1.2 definierten Fehlermeldungen.

Darüber hinaus gibt es keine spezifischen Fehlermeldungen für diesen Endpunkt.

### 10.2.2 [/organisationen/{id}] Lesen einer individuellen Organisation

Die Schnittstelle /v1/organisationen/{organisation.id} bezieht sich auf das Anfordern einer Organisation per ID.

https://subdomain.domain.org/subpaths/v1/organisationen/{organisation.id}

Quellcode 99: URL-Struktur des Endpunkts organisationen/{organisation.id}

Der Pfad-Parameter {organisation.id} bezieht sich auf die ID der Organisation.

Bei Zugriffen auf den Endpunkt ist folgende Operation und HTTP-Methode erlaubt:

Operation	HTTP-Methode
READ	GET

Tabelle 28: Zugelassene Operation auf dem Endpunkt /v1/organisationen/{organisation.id}



## READ

Dieser Schnittstellenendpunkt gibt die Organisation zur angegebenen `{organisation.id}` zurück.

Ein READ ist nur mit HTTP-GET auf die API

`/v1/organisationen/{organisation.id}` zum Abfragen einer Organisation per ID zugelassen. Die Antwort-Nutzdaten (Response Payload) beinhalten ein JSON-Objekt des Datentyps Organisation. Das Datenmodell „Organisation“ ist in 5.2 beschrieben.

### Request

Die Anfrage erfordert keine Nutzdaten (Request Payload).

### Response

Nachfolgend die JSON-Struktur der Antwort-Nutzdaten (Response Payload):

```
{
  "id": <String>, // ID der Organisation
  "kennung": <String>, // Kennung der Organisation
  "name": <String>, // Name der Organisation
  "namensergaenzung": <String>, // Namensergänzung der Organisation
  "kuerzel": <String>, // Kürzel der Organisation
  "anschrift": <Anschrift>, // Anschrift der Organisation
  "typ": <Code>, // Siehe Code-Liste "Organisation / Typ"
  "traegerschaft": <Code>, // Siehe Code-Liste "Trägerschaft"
} // Siehe "Organisation"
```

*Quellcode 100: JSON-Struktur einer READ-Response an `/v1/organisationen/{id}`*

Beispiel Antwort-Nutzdaten (Response Payload):

```
{
  "id": "c7de2c10-26a6-4d12-8152-f9e8db497625",
  "kennung": "NI_12346",
  "name": "Muster-Schule",
  "anschrift" : {
    "postleitzahl": "30519",
    "ort": "Hannover",
    "ortsteil": "Döhren"
  },
}
```

```

    "typ": "SCHULE",
    "traegerschaft": "02"
  }

```

Quellcode 101: Beispiel Response zu /organisationen/{organisation.id} auf HTTP-GET

## Mögliche Fehlermeldungen

Zugrunde liegen die in 7.1.2 definierten Fehlermeldungen.

Darüber hinaus gibt es keine spezifischen Fehlermeldungen für diesen Endpunkt.

### 10.2.3 [/organisationen/{id}/organisationsbeziehungen] Lesen von zu einer Organisation gehörenden Organisationsbeziehungen

Die Schnittstelle /v1/organisationen/{id}/organisationsbeziehungen erlaubt das Lesen von zu einer Organisation gehörenden Organisationsbeziehungen.

Organisationsbeziehungen sind immer von der Form "hat als". Besteht von einer Organisation A eine Organisationsbeziehung vom Typ "SchBeh" zu einer Organisation B, so bedeutet dieses, das Organisation A die Organisation B als Schulbehörde hat.

Bestehen zwischen zwei Organisationen mehrere Beziehungen (ist beispielsweise eine Organisation sowohl Schulbehörde als auch Schulträger einer anderen Organisation), so hat jede Beziehung ein eigenes Organisationsbeziehungsobjekt.

```

https://subdomain.domain.org/subpaths/v1/organisationen/{id}/organisationsbeziehungen

```

Quellcode 102: URL-Struktur des Endpunkts organisationen/{id}/organisationsbeziehungen

Auf Organisationsbeziehungen kann nur lesend zugegriffen werden.

Bei Zugriffen auf den Endpunkt ist folgende Operation und HTTP-Methode erlaubt:

Operation	HTTP-Methode
READ	GET

Tabelle 29: Zugelassene Operation auf dem Endpunkt /v1/organisationen/{id}/organisationsbeziehungen

## READ

Dieser Schnittstellenendpunkt gibt die Organisationsbeziehungen zur angegebenen Organisations-ID zurück.

Ein READ ist nur mit HTTP-GET auf die API `/v1/organisationen/{id}/organisationsbeziehungen` zum Abfragen der Organisationsbeziehungen per Organisations-ID zugelassen. Die Antwort-Nutzdaten (Response Payload) beinhalten ein JSON-Objekt mit einem Array des Datentyps Organisationsbeziehung. Das Datenmodell „Organisationsbeziehung“ ist in 5.13 beschrieben.

Die Schnittstelle `/v1/organisationen/{id}/organisationsbeziehungen` ermöglicht das Verwenden von Filterparametern zur Präzisierung der Anfrage bei HTTP-GET.

Folgende Tabelle listet die zur Verfügung stehenden Filter auf:

Filter	Typ	Beschreibung
hat_als_organisationsbeziehungen	String (Boolean)	listet alle Organisationsbeziehungen auf, welche die, durch die Organisations-ID gegebene, Organisation hat  Die Organisationsbeziehungen werden unter 'hat_als_organisationsbeziehungen' aufgelistet.  Der Defaultwert ist "ja", es werden also ohne Filterangabe alle Organisationsbeziehungen angegeben, welche eine Organisation hat.
ist_von_organisationsbeziehungen	String (Boolean)	listet alle Organisationsbeziehungen auf, welche von anderen Organisationen zu dem, durch die Organisations-ID gegebenen, Organisation bestehen  Die Organisationsbeziehungen werden unter 'ist_von_organisationsbeziehungen' aufgelistet.  Der Defaultwert ist "nein", es werden also ohne Filterangabe keine Organisationsbeziehungen angegeben, welche aus der gegebenen Organisation heraus bestehen.

*Tabelle 30: Zugelassene Filter auf dem Endpunkt `/v1/organisationen/{id}/organisationsbeziehungen`*

## Request

Die Anfrage erfordert keine Nutzdaten (Request Payload).

## Response

Nachfolgend die JSON-Struktur der Antwort-Nutzdaten (Response Payload):

```
{
  "hat_als_organisationsbeziehungen": [ // Array von
Organisationsbeziehungen, welche von der Organisation ausgehen
    {
      "orgid": <Organisations-ID auf welche die
Organisationsbeziehung referenziert >
      "organisationsbeziehung": <Art der Beziehung>
    },
    ...
  ],
  "ist_von_organisationsbeziehungen ": [
    {
      "orgid": <Organisations-ID von welcher eine
Organisationsbeziehung aus referenziert >
      "organisationsbeziehung": <Art der Beziehung>
    }
  ],
}
```

*Quellcode 103: JSON-Struktur einer READ-Response an /v1/organisationen/{id}/organisationsbeziehungen*

Beispiel Antwort-Nutzdaten (Response Payload):

```
{
  "hat_als_organisationsbeziehungen": [
    {
      "orgid": "692efeee-7587-4f9d-9134-2282cea45fd4",
      "organisationsbeziehung": "SchTrae"
    },
    {
      "orgid": "131feda8-5ed5-4e5f-be85-10b0247d2e9e",
      "organisationsbeziehung": "SchBeh"
    }
  ]
}
```

```

    }
  ],
}

```

Quellcode 104: Beispiel Response zu `/v1/organisationen/{id}/organisationsbeziehungen` auf HTTP-GET ohne Filter

### Beispiel Antwort-Nutzdaten (Response Payload):

```

{
  "ist_von_organisationsbeziehungen": [
    {
      "orgid": "6359b712-a382-4931-97b0-a17092aa66d9",
      "organisationsbeziehung": "Schbeh"
    },
    {
      "orgid": "c27b01f7-88cf-46e7-b686-108564b546fd",
      "organisationsbeziehung": "Schbeh"
    },
    {
      "orgid": "0fef816a-fe85-49ad-8847-3810b6cad7b7",
      "organisationsbeziehung": "Schbeh"
    },
    {
      "orgid": "6cdfac42-21aa-4b95-8ea9-55db419e8bca",
      "organisationsbeziehung": "Schbeh"
    },
    {
      "orgid": "10322664-4726-453c-bbec-4064d10f3777",
      "organisationsbeziehung": "Schbeh"
    },
    {
      "orgid": "206be715-8477-4bde-8494-fb928bd79984",
      "organisationsbeziehung": "Schbeh"
    },
    {
      "orgid": "0fel1f58c-e2b8-43a3-b34e-1fba47233be7",
      "organisationsbeziehung": "Schbeh"
    }
  ],
}

```

```
{
  "orgid": "0483f8fc-b625-4a18-890e-7589ea0d3ba2",
  "organisationsbeziehung": "Schbeh"
},
]
```

*Quellcode 105: Beispiel Response zu /v1/organisationen/{id}/organisationsbeziehungen auf HTTP-GET mit Filter  
"?hat\_als\_organisationsbeziehungen=nein&ist\_von\_organisationsbeziehungen=ja"*

### Mögliche Fehlermeldungen

Zugrunde liegen die in 7.1.2 definierten Fehlermeldungen.

Darüber hinaus gibt es keine spezifischen Fehlermeldungen für diesen Endpunkt.

#### 10.2.4 [/organisation-info] Lesen der eigenen Organisation

Die Schnittstelle /v1/organisation-info bezieht sich auf das Anfordern von Informationen zu der Organisation des Quellsystems. Die benötigten Daten werden bei der Anfrage mit dem access\_token übermittelt.

<code>https://subdomain.domain.org/subpaths/v1/organisation-info</code>
---

*Quellcode 106: URL-Struktur des Endpunkts organisation-info*

Bei Zugriffen auf den Endpunkt ist folgende Operation und HTTP-Methode erlaubt:

Operation	HTTP-Methode
READ	GET

*Tabelle 31: Zugelassene Operation auf dem Endpunkt /v1/organisation-info*

## READ

Dieser Schnittstellenendpunkt gibt die Informationen der anfordernden Organisation zurück.

Ein READ ist nur mit HTTP-GET auf die API `/v1/organisation-info` zum Abfragen einer Information zu einer Organisation zugelassen. Die Antwort-Nutzdaten (Response Payload) beinhalten ein JSON-Objekt des Datentyps Organisation. Das Datenmodell „Organisation“ ist in 5.2 beschrieben.

### Request

Die Anfrage erfordert keine Nutzdaten (Request Payload).

### Response

Nachfolgend die JSON-Struktur der Antwort-Nutzdaten (Response Payload):

```
{
  "id": <String>, // ID der Organisation
  "kennung": <String>, // Kennung der Organisation
  "name": <String>, // Name der Organisation
  "namensergaenzung": <String>, // Namensergänzung der Organisation
  "kuerzel": <String>, // Kürzel der Organisation
  "anschrift": <Anschrift>, // Anschrift der Organisation
  "typ": <Code>, // Siehe Code-Liste "Organisation / Typ"
  "traegerschaft": <Code>, // Siehe Code-Liste "Trägerschaft"
} // Siehe "Organisation"
```

*Quellcode 107: JSON-Struktur einer READ-Response an `/v1/organisation-info`*

Beispiel Antwort Nutzdaten (Response Payload):

```
{
  "id": "c7de2c10-26a6-4d12-8152-f9e8db497625",
  "kennung": "NI_12-34-56",
  "name": "Muster-Schule",
  "anschrift" : {
    "postleitzahl": "30519",
    "ort": "Hannover",
    "ortsteil": "Döhren"
  },
}
```



```
    "typ": "SCHULE",  
    "traegerschaft": "02"  
}
```

*Quellcode 108: Beispiel Response zu /organisation-info auf HTTP-GET*

### Mögliche Fehlermeldungen

Zugrunde liegen die in 7.1.2 definierten Fehlermeldungen.

Darüber hinaus gibt es keine spezifischen Fehlermeldungen für diesen Endpunkt.

## 10.3 Schnittstellen für Gruppen

### 10.3.1 [/gruppen] Gruppen erstellen und lesen

Die Schnittstelle /v1/gruppen bezieht sich auf die Auflistung von Gruppen.

<code>https://subdomain.domain.org/subpaths/v1/gruppen</code>
---

Quellcode 109: URL-Struktur des Endpunkts Gruppen

Bei Zugriffen auf den Endpunkt sind folgende Operationen und HTTP-Methoden erlaubt:

Operation	HTTP-Methode
CREATE	POST
READ	GET

*Tabelle 32: Zugelassene Operationen auf dem Endpunkt /v1/gruppen*

## CREATE

Mittels dieser Schnittstelle wird ein neuer Datensatz "Gruppe" erstellt.

Ein CREATE ist nur mit HTTP-POST auf die API /v1/gruppen zum Erstellen von Gruppen zugelassen. Die Anfrage-Nutzdaten (Request Payload) beinhalten ein JSON-Objekt des Datentyps Gruppe. Das Datenmodell „Gruppe“ ist in 5.6 beschrieben. Die Gruppen-ID, die Mandanten-ID, die Organisations-ID und die Revisionsnummer werden vom SchulConneX-Service vergeben und hinzugefügt.

### Request

Die JSON-Struktur für das Erstellen einer Gruppe ergibt sich wie folgt (Request Payload):

```
{
  "referrer": <String>, // Organisationseigene ID der Gruppe
  "bezeichnung": <String>, //Bezeichnung der Gruppe
  "thema": <String>, //Optionales Thema der Gruppe
  "beschreibung": <String>, //Optionale Beschreibung der Gruppe
  "typ": <Code>, // Siehe Code-Liste Gruppentyp
  "bereich": <Code>, // Siehe Code-Liste Gruppen
  "optionen": [<Code>, <Code>, <Code>...],
  // Siehe Code-Liste Gruppenoption
  "differenzierung": <Code>,
  // Siehe Code-Liste Gruppendifferenzierung
  "bildungsziele": [<Code>, <Code>, <Code>...]
  // Siehe Code-Liste Bildungsziel
  "jahrgangsstufen": [<Code>, <Code>, <Code>...]
  // Siehe Code-Liste Jahrgangsstufe
  "faecher": [<faecher>, <faecher>, <faecher>...]
  // Objekte mit Faecher Attributen, siehe Beschreibung "Gruppe"
  "referenzgruppen": [{
    "gruppid": <String>, // ID der referenzierten Gruppe
    "rollen": [<Code>, <Code>, <Code>...]
    // Aus der Referenzgruppe zu übernehmende Gruppenrollen -
    // siehe Code-Liste Gruppenrollen
  }]
  "laufzeit": {
    // von/vonlernperiode und bis/bislernperiode sind jeweils alternativ
    "von": <Datum>, // Anfangsdatum
    "vonlernperiode": <Code>, // Referenz auf Codeliste Lernperiode
    "bis": <Datum>, // Enddatum
    "bislernperiode": <Code>, // Referenz auf Codeliste Lernperiode
  }
}
```

```
}  
}
```

*Quellcode 110: JSON-Struktur eines CREATE-Requests an /v1/gruppen*

## Response

Bei einer erfolgreichen Anforderung zum Erstellen einer Gruppe wird diese Anforderung mit einer Repräsentation der Gruppe in den Antwort-Nutzdaten (Response Payload) und dem HTTPStatusCode 200 („200 OK“) quittiert.

Nachfolgend die JSON-Struktur der Antwort-Nutzdaten (Response Payload):

```
<Gruppe> // Siehe "Gruppe"
```

*Quellcode 111: JSON-Struktur einer CREATE-Response an /v1/gruppen*

Beispiel Antwort-Nutzdaten (Response Payload):

```
{  
  "id": "c7de2c10-26a6-4d12-8152-f9e8db497625",  
  "mandant": "9b3f36ad-9d15-49f9-9660-6cf9746ba446",  
  "referer": "HHG-1281b688-d639",  
  "bezeichnung": "Englischkurs Klasse 3b",  
  "typ": "KURS",  
  "bereich": "PFLICHT",  
  "optionen": ["01"],  
  "differenzierung": "G",  
  "bildungsziele": ["RS"],  
  "jahrgangsstufen": ["03"],  
  "faecher": [{  
    "kennung": "EN"  
  }],  
  "referenzgruppen": [{  
    "grupid": "e70663cb-65c4-4ad5-a9f1-7edbc3c36c4b",  
    "rollen": ["Lern", "GMit"]  
  }],  
  {
```

```
        "id": "11e84d3f-d78e-43b7-8c20-d26e43eb3761",
        "rollen": ["Lern"]
    }
],
"laufzeit": {
    "von": "2022-08-01",
    "bis": "2023-07-31"
},
"revision": "1"
}
```

*Quellcode 112: Beispiel Response zu /gruppen auf HTTP-POST*

### Mögliche Fehlermeldungen

Zugrunde liegen die in 7.1.2 definierten Fehlermeldungen.

Darüber hinaus gibt es keine spezifischen Fehlermeldungen für diesen Endpunkt.

## READ

Dieser Schnittstellenendpunkt gibt ein Array von Gruppendatensätzen zurück, die vom Anfrager ausgelesen werden dürfen.

Ein READ kann nur mit einer HTTP-GET Anfrage erfolgen. Die Antwort-Nutzdaten (Response Payload) beinhalten ein Array von JSON-Objekten vom Datentyp Gruppendatensatz, sofern Einträge existieren. Das Datenmodell „Gruppendatensatz“ ist in 5.8 beschrieben.

Die Schnittstelle /v1/gruppen ermöglicht das Verwenden von Filterparametern zur Präzisierung der Anfrage bei HTTP-GET.

Durch das Setzen von Filterparametern (query-parameter) wird mit der HTTP-GET-Anfrage eine Filteroperation mitgeschickt, welche die Ergebnismenge der Antwort-Nutzdaten (Response Payload) beeinflusst, beispielsweise mit den Filterparametern „?referrer=123&faecher=EN“.

Bei Filtern vom Typ "String" überprüft der SchulConneX-Server, ob der Wert des Filterparameters ohne Beachtung von Groß- und Kleinschreibung im Filter-Attribut beinhaltet ist (contains). Bei Filtern vom Typ "Code" überprüft der SchulConneX-Server, ob der Wert des Filterparameters ohne Beachtung von Groß- und Kleinschreibung gleich dem Filter-Attribut ist (equals).

Werden mehrere Filter angegeben, so sind sie als mit 'und' verknüpft anzusehen.

Die mehrfache Nutzung des gleichen Filters in einer Anfrage ist nicht zulässig.

Bei Attributen, welche eine Liste von Codes enthalten können (, deren 'Anzahl' im Datenmodell die Form "0..n" oder "1..n" hat), kann als Filterwert eine durch Komma getrennte Liste von Werten angegeben werden. Es werden dann nur die Gruppen zurück geliefert, in denen alle diese Codes in der Liste vorkommen. Die Parameter sind auch hier als mit 'und' verknüpft anzusehen. Der Filter "&faecher=NL, IT, SN" liefert alle Gruppen, in denen kumulativ sowohl Niederländisch, Italienisch und Spanisch gelehrt wird.

Ein Sonderfall ist hier bei dem Filter "faecher" gegeben. Ein Suchparameter bei diesem Filter sucht sowohl bei dem Unterattribut "kennung" als auch dem Unterattribut "text". Wird in einer Gruppe beispielsweise das Standard-Fach Englisch unterrichtet und auch das Thema "Filmschnitt", das nicht in der Fächer-Codeliste vorhanden ist und daher über das Attribut "Text" hinzugefügt wurde, so kann diese Gruppe mit dem Filter "&faecher=EN, Filmschnitt" gesucht werden.

Folgende Tabelle listet die zur Verfügung stehenden Filter:

Filter	Typ	Beschreibung
referrer	String	Möglichkeit, die Ergebnisliste der Anfrage nach dem Fremdschlüssel "referrer" zu filtern
bezeichnung	String	Möglichkeit, die Ergebnisliste der Anfrage nach dem Attribut "gruppe.bezeichnung" zu filtern
optionen	Code	Möglichkeit, die Ergebnisliste der Anfrage nach dem Attribut "gruppe.optionen" zu filtern
differenzierung	Code	Möglichkeit, die Ergebnisliste der Anfrage nach dem Attribut "gruppe.differenzierung" zu filtern
bildungsziele	Code	Möglichkeit, die Ergebnisliste der Anfrage nach dem Attribut "gruppe.bildungsziele" zu filtern
jahrgangsstufen	Code	Möglichkeit, die Ergebnisliste der Anfrage nach dem Attribut "gruppe.jahrgangsstufen" zu filtern
faecher	Code	Möglichkeit, die Ergebnisliste der Anfrage nach dem Attribut "gruppe.faecher.kennung" und "gruppe.faecher.text" zu filtern
sichtfreigabe	String	<p>erlaubt es Gruppen, abhängig davon aufzulisten, ob diese von einer anderen Organisation zum Lesen freigegeben wurden</p> <p>Die möglichen Optionen sind "sichtfreigabe=ja", um nur die durch externe Organisationen freigegebenen Gruppen aufzulisten, und "sichtfreigabe=nein", um nur eigene Gruppen aufzulisten. Wird nicht nach Sichtfreigabe gefiltert, so werden alle lesbaren Gruppen geliefert.</p>

Tabelle 33: Zugelassene Filter auf dem Endpunkt /v1/gruppen

### Request

Die Anfrage erfordert keine Nutzdaten (Request Payload).

### Response

Nachfolgend die JSON-Struktur der Antwort-Nutzdaten (Response Payload):

```
[
  {
    "gruppe": <Gruppe>, // Siehe "Gruppe".
    "gruppenzugehoerigkeiten": [
      <Gruppenzugehoerigkeit> // Siehe "Gruppenzugehoerigkeit".
    ]
  }
]
```

Quellcode 113: JSON-Struktur eines READ-Requests an /v1/personen

Nachfolgend die JSON-Struktur der Antwort-Nutzdaten (Response Payload):

```
[
  {
    "gruppe": {
      "id": <String>, // ID der Gruppe
      "mandant": <String>, // ID des Mandanten.
      "orgid": <String>, // ID des Organisation
      "referrer": <String>, // Organisationseigene ID der Gruppe
      "bezeichnung": <String>, //Bezeichnung der Gruppe
      "thema": <String>, //Optionales Thema der Gruppe
      "beschreibung": <String>, //Optionale Beschreibung der Gruppe
      "typ": <Code>, // Siehe Code-Liste Gruppentyp
      "bereich": <Code>, // Siehe Code-Liste Gruppen
      "optionen": [<Code>, <Code>, <Code>...], // Siehe Code-Liste
      Gruppenoption
      "differenzierung": <Code>, // Siehe Code-Liste
      Gruppendifferenzierung
      "bildungsziele": [<Code>, <Code>, <Code>...] // Siehe Code-Liste
      Bildungsziel
      "jahrgangsstufen": [<Code>, <Code>, <Code>...] // Siehe Code-Liste
      Jahrgangsstufe
      "faecher": [<faecher>, <faecher>, <faecher>...] // Objekte mit
      Faecher Attributen, siehe Beschreibung "Gruppe"
      "referenzgruppen": [{
        "grupid": <String>, // ID der referenzierten Gruppe
        "rollen": [<Code>, <Code>, <Code>...] // Aus der Referenzgruppe
        zu übernehmende Gruppenrollen - siehe Code-Liste Gruppenrollen
      }]
    }
  }
]
```



```

    "laufzeit": {
        "von": <Datum>, // Anfangsdatum
        "bis": <Datum>, // Enddatum
    },
    "revision": <String> // revisionsnummer
},
    "gruppenzugehoerigkeiten": [
        {
            "id": <String>, // ID der Gruppenzugehoerigkeit
            "ktid": <String>, // ID des Personenkontexts
            "rollen": [<Code>, <Code>, <Code>...], // Rolle(n) in
der Gruppe
            "von": <Datum>, // Anfang der Gruppenzugehoerigkeit
            "bis": <Datum>, // Ende der Gruppenzugehoerigkeit
            "revision": <String> // revisionsnummer
        }, ...
    ]
}
]

```

Quellcode 114: JSON-Struktur einer READ-Response an /v1/Gruppe

#### Beispiel Antwort-Nutzdaten (Response Payload):

```

[ {
    "gruppe": {
        "id": "c7de2c10-26a6-4d12-8152-f9e8db497625",
        "mandant": "9b3f36ad-9d15-49f9-9660-6cf9746ba446",
        "orgid": "9b3f36ad-9d15-49f9-9660-6cf9746ba446",

        "referer": "HHG-1281b688-d639",
        "bezeichnung": "Englischkurs Klasse 3b",
        "typ": "KURS",
        "bereich": "PFLICHT",
        "optionen": ["01"],
        "differenzierung": "G",
        "bildungsziele": ["RS"],
    }
}
]

```

```

        "jahrgangsstufen": ["03"],
        "faecher": [{
            "kennung": "EN"
        }],
        "laufzeit": {
            "von": "2022-08-01",
            "bis": "2023-07-31"
        },
        "revision": "1"
    },
    "gruppenzugehoerigkeiten": [{
        "id": "c85af1ac-ddaa-4a77-ba74-1836682b9afa",
        "referrer": "c85af1ac-ddaa-ba74-1836682bacac",
        "ktid": "23643c9e-956a-4baa-af4e-813e0a7efa4b",
        "rollen": ["LERN"],
        "von": "2022-08-01",
        "bis": "2023-07-31"
    },
    {
        "id": "b5c17b1c-ee96-4132-9dac-5b165459b972",
        "mandant": "de45f1ac-ddaa-4a77-ba74-1836682b86de",
        "referrer": "c85af1ac-ddaa-ba74-1836682bacdd",
        "ktid": "01a51971-6239-4f8a-8973-e38f5cf21881",
        "rollen": ["LEHR"],
        "von": "2022-08-01",
        "bis": "2023-07-31"
    }
    ]
},
{
    "Gruppe": {
        "id": "c7de2c10-26a6-4d12-8152-f9e8db625497",
        "mandant": "9b3f36ad-9d15-49f9-9660-6cf9746ba446",
        "orgid": "11600ce6-a970-4230-9d84-c38bb51ac20d",
        "referer": "HHG-1281b688-d6732",
        "bezeichnung": "Mathe Klasse 4a",
    }
}

```

```

        "typ": "KLASSE",
        "bereich": "PFLICHT",
        "optionen": ["02"],
        "differenzierung": "G",
        "bildungsziele": ["RS"],
        "jahrgangsstufen": ["04"],
        "faecher": [{
            "kennung": "EN"
        },
        {
            "kennung": "EK"
        },
        {
            "text": "Wassersport"
        }
    ],
    "laufzeit": {
        "von": "2022-08-01",
        "bis": "2023-07-31"
    },
    "revision": "1"
},
"gruppenzugehoerigkeiten": [{
    "id": "8722b058-8747-4646-8da0-d523a9f619b8",
    "referrer": "c85af1ac-ddaa-ba74-1836682bdada",
    "ktid": "42865c3f-2806-4488-9d46-dbaeb004bc8e",
    "rollen": ["LERN"],
    "von": "2022-08-01",
    "bis": "2023-07-31",
    "revision": "1"
},
{
    "id": " ed4a1432-6a58-4e99-b89e-e1d82b0052de",
    "mandant": "27431ac-ddaa-4a77-ba74-1836682b5634",
    "referrer": "c85af1ac-ddaa-ba74-1836682bcfcf",

```

```
    "ktid": "dc8bfbb0-a6ab-434a-a52c-bde200385d97",  
    "rollen": ["LERN"],  
    "von": "2022-08-01",  
    "bis": "2023-07-31",  
    "revision": "1"  
  }  
]  
}
```

*Quellcode 115: Beispiel Response zu /gruppen auf HTTP-GET*

### Mögliche Fehlermeldungen

Zugrunde liegen die in 7.1.2. definierten Fehlermeldungen.

Darüber hinaus gibt es keine spezifischen Fehlermeldungen für diesen Endpunkt.

### 10.3.2 [/gruppen/{id}] Lesen, ändern und löschen einer individuellen Gruppe

Die Schnittstelle /v1/gruppen/{gruppe.id} bezieht sich auf das Anfordern einer Gruppe per ID.

`https://subdomain.domain.org/subpaths/v1/gruppen/{gruppe.id}`

*Quellcode 116: URL-Struktur des Endpunkts gruppen/{gruppe.id}*

Der Pfad-Parameter {gruppe.id} bezieht sich auf die ID der Gruppe.

Bei Zugriffen auf den Endpunkt sind folgende Operationen und HTTP-Methoden erlaubt:

Operation	HTTP-Methode
READ	GET
UPDATE	PUT
DELETE	DELETE

*Tabelle 34: Zugelassene Operationen auf dem Endpunkt /v1/gruppen/{gruppe.id}*

## READ

Dieser Schnittstellenendpunkt gibt die Gruppe zur angegebenen `{gruppe.id}` zurück.

Ein READ ist nur mit HTTP-GET auf die API `/v1/gruppen/{gruppe.id}` zum Abfragen einer Gruppe per ID zugelassen. Die Antwort-Nutzdaten (Response Payload) beinhalten ein JSON-Objekt des Datentyps Gruppendatensatz. Das Datenmodell „Gruppendatensatz“ ist in 5.8 beschrieben.

### Request

Die Anfrage erfordert keine Nutzdaten (Request Payload).

### Response

Nachfolgend die JSON-Struktur der Antwort-Nutzdaten (Response Payload):

```
{
  "gruppe": <Gruppe>, // Siehe "Gruppe"
  "gruppenzugehoerigkeiten": [
    <Gruppenzugehoerigkeit> // Siehe Gruppenzugehoerigkeit
  ]
}
```

*Quellcode 117: JSON-Struktur der Antwort bei READ auf `/gruppen/{gruppe.id}`*

Beispiel Antwort-Nutzdaten (Response Payload):

```
{
  "gruppe": {
    "id": "57453f9b-2024-4a94-afb5-72d832a32903",
    "mandant": "9b3f36ad-9d15-49f9-9660-6cf9746ba446",
    "orgid": "9b3f36ad-9d15-49f9-9660-6cf9746ba446",
    "referer": "HHG-1281b688-d639",
    "bezeichnung": "Englischkurs Klasse 3b",
    "typ": "KURS",
    "bereich": "PFLICHT",
    "optionen": ["01"],
    "differenzierung": "G",
    "bildungsziele": ["RS"],
    "jahrgangsstufen": ["03"],
    "faecher": [{
```

```

        "kennung": "EN"
    }],
    "referenzgruppen": [{
        "grupid": "e70663cb-65c4-4ad5-a9f1-7edbc3c36c4b",
        "rollen": ["Lern", "GMit"]
    },
    {
        "grupid": "11e84d3f-d78e-43b7-8c20-d26e43eb3761",
        "rollen": ["Lern"]
    }
    ],
    "laufzeit": {
        "von": "2022-08-01",
        "bis": "2023-07-31"
    },
    "revision": "1"
},
"groupenzugehoerigkeiten": [{
    "id": "8722b058-8747-4646-8da0-d523a9f619b8",
    "ktid": "42865c3f-2806-4488-9d46-dbaeb004bc8e",
    "rollen": ["LERN"],
    "von": "2022-08-01",
    "bis": "2023-07-31",
    "revision": "1"
},
{
    "id": " ed4a1432-6a58-4e99-b89e-e1d82b0052de",
    "mandant": " c4e869-774a-42a4-8537-b8151c26dde0",
    "ktid": "dc8bfbb0-a6ab-434a-a52c-bde200385d97",
    "rollen": ["LERN"],
    "von": "2022-08-01",
    "bis": "2023-07-31",
    "revision": "1"
}
]
}

```

Quellcode 118: Beispiel Response zu /gruppen/{gruppe.id} auf HTTP-GET

### Mögliche Fehlermeldungen

Zugrunde liegen die in 7.1.2. definierten Fehlermeldungen.

Darüber hinaus gibt es keine spezifischen Fehlermeldungen für diesen Endpunkt.



## UPDATE

Dieser Schnittstellenendpunkt aktualisiert die Gruppe zur angegebenen `{gruppen.id}`.

Ein UPDATE ist nur mit HTTP-PUT auf die API `/v1/gruppen/{gruppen.id}` zum Aktualisieren einer Gruppe per ID zugelassen. Die Anfrage-Nutzdaten (Request Payload) beinhalten ein JSON-Objekt des Datentyps Gruppe. Das Datenmodell „Gruppe“ ist in 5.6 beschrieben.

Es ist erforderlich, dass zur Aktualisierungsanfrage in der bearbeiteten Gruppe das Attribut „gruppe.revision“ der zugrunde liegenden Gruppe mitgeschickt wird. Der SchulConneX-Server überprüft anhand des mitgeschickten Wertes des Attributs „gruppe.revision“, ob der Datensatz der Gruppe in der Zwischenzeit keine Änderung erfahren hat. Der Wert des Attributs kann von einem Quellsystem oder Dienst nicht verändert werden; dieser wird mit der Erstellung (Create) des Datensatzes Gruppe und jeder erfolgreichen Änderung (Update) vom SchulConneX-Server neu generiert.

Die folgende Tabelle listet die Attribute einer Gruppe auf, welche von einem Quellsystem oder Dienst nicht verändert werden können.

Attribut	Bemerkung
id	ID der Gruppe wird vom SchulConneX-Server vergeben und ist eindeutig Dieses Attribut ist unveränderbar (immutable).
mandant	Mandanten-ID der Gruppe wird vom SchulConneX-Server vergeben und ist eindeutig Dieses Attribut ist unveränderbar (immutable).
orgid	Organisations-ID der Gruppe wird automatisch vergeben und ist eindeutig Dieses Attribut ist unveränderbar (immutable).
sichtfreigabe	gibt an, ob diese Gruppe nur aufgrund einer Freigabe durch eine andere Organisation sichtbar ist  Sollte der Wert von Sichtfreigabe "ja" sein, führt ein Update auf diese Gruppe zu einer Fehlermeldung, da eine Sichtfreigabe nur das Lesen der Gruppe erlaubt.
revision	Revision der Gruppe

	<p>wird vom SchulConneX-Server mit der Erstellung des Datensatzes sowie Aktualisierung generiert</p> <p>Dieser Wert kann nicht von Quellsystemen oder Diensten gesetzt werden.</p>
--	--

*Tabelle 35: Liste von Attributen, welche durch ein Quellsystem oder Dienst über den Endpunkt /v1/gruppen/{gruppe.id} auf HTTP-PUT nicht verändert werden können.*

Die Attribute id, mandant und orgid können im Request ausgelassen werden. Sind ein oder mehrere Attribute im Request dennoch vorhanden, wird es oder werden sie validiert und führen gegebenenfalls zu einer Fehlermeldung.

## Request

In Anbetracht der Attribute, die von einem Quellsystem oder Dienst bei der Aktualisierung einer Gruppe nicht gesetzt oder verändert werden können, ergibt sich folgende JSON-Struktur der Anfrage-Nutzdaten (Request Payload):

```
{
  "id": <String>, // ID der Gruppe
  "mandant": <String>, // ID des Mandanten.
  "orgid": <String>, // ID der Organisation
  "referrer": <String>, // Organisationseigene ID der Gruppe
  "bezeichnung": <String>, //Bezeichnung der Gruppe
  "thema": <String>, //Optionales Thema der Gruppe
  "beschreibung": <String>, //Optionale Beschreibung der Gruppe
  "typ": <Code>, // Siehe Code-Liste
  "bereich": <Code>, // Siehe Code-Liste
  "optionen": [<Code>, <Code>, <Code>...], // Siehe Code-Liste
  "differenzierung": <Code>, // Siehe Code-Liste
  "bildungsziele": [<Code>, <Code>, <Code>...] // Siehe Code-Liste
  "jahrgangsstufen": [<Code>, <Code>, <Code>...] // Siehe Code-Liste
  "faecher": [<faecher>, <faecher>, <faecher>...]
  // Objekte mit Faecher Attributen,siehe Beschreibung "Gruppe"
  "referenzgruppen": [{
    "grupid": <String>, // ID der referenzierten Gruppe
    "rollen": [<Code>, <Code>, <Code>...]
    // Aus der Referenzgruppe zu übernehmende Gruppenrollen -
    // siehe Code-Liste Gruppenrollen
  }]
  "laufzeit": {
    // von/vonlernperiode und bis/bislernperiode sind
    // jeweils alternativ
  }
}
```

```

    "von": <Datum>, // Anfangsdatum
    "vonlernperiode": <Code>, // Referenz auf Codeliste Lernperiode
    "bis": <Datum>, // Enddatum
    "bislernperiode": <Code>, // Referenz auf Codeliste Lernperiode
  },
  "revision": <String>, // Revisionsnummer
}

```

*Quellcode 119: JSON-Struktur eines UPDATE-Requests an /v1/gruppen/{gruppen.id}*

Anmerkung: Bei dieser Aktualisierungsanfrage handelt es sich um eine HTTP-PUT-Operation, bei der die Anfrage-Nutzlast mit den gegebenen Attributen der Aktualisierung zugrunde gelegt wird (mitgeschickte Attribute werden aktualisiert, nicht mitgeschickte Attribute werden gelöscht); es ist keine HTTP-PATCH-Operation, bei der lediglich die zu aktualisierenden Attribute in der Anfrage-Nutzlast mitgeschickt werden.

## Response

Bei einer erfolgreichen Anforderung zum Aktualisieren einer Gruppe wird diese Anforderung mit einer Repräsentation der Gruppe in den Antwort-Nutzdaten (Response Payload) und dem HTTP Status Code 200 („200 OK“) quittiert.

Nachfolgend die JSON-Struktur der Antwort-Nutzdaten (Response Payload):

```

{
  "id": <String>, // ID der Gruppe
  "mandant": <String>, // ID des Mandanten.
  "orgid": <String>, // ID der Organisation.
  "referrer": <String>, // Organisationseigene ID der Gruppe
  "bezeichnung": <String>, //Bezeichnung der Gruppe
  "thema": <String>, //Optionales Thema der Gruppe
  "beschreibung": <String>, //Optionale Beschreibung der Gruppe
  "typ": <Code>, // Siehe Code-Liste
  "bereich": <Code>, // Siehe Code-Liste
  "option": [<Code>, <Code>, <Code>...], // Siehe Code-Liste
  "differenzierung": <Code>, // Siehe Code-Liste
  "bildungsziele": [<Code>, <Code>, <Code>...] // Siehe Code-Liste
  "jahrgangsstufen": [<Code>, <Code>, <Code>...] // Siehe Code-Liste
  "faecher": [<faecher>, <faecher>, <faecher>...] // Objekte mit
  Faecher Attributen,siehe Beschreibung "Gruppe"
  "referenzgruppen": [{
    "grupid": <String>, // ID der referenzierten Gruppe

```

```

        "rollen": [<Code>, <Code>, <Code>...] // Aus der Referenzgruppe
zu übernehmende Gruppenrollen - siehe Code-Liste Gruppenrollen

    ]]
    "laufzeit": { // von/vonlernperiode und bis/bislernperiode sind
jeweils alternativ
        "von": <Datum>, // Anfangsdatum
        "vonlernperiode": <Code>, // Referenz auf Codeliste Lernperiode
        "bis": <Datum>, // Enddatum
        "bislernperiode": <Code>, // Referenz auf Codeliste Lernperiode
    },
    "revision": <String>, // Revisionsnummer
}

```

*Quellcode 120: JSON-Struktur einer UPDATE-Response an /v1/gruppen/{gruppe.id}*

#### Beispiel Antwort-Nutzdaten (Response Payload):

```

{
    "id": "c7de2c10-26a6-4d12-8152-f9e8db497625",
    "mandant": "9b3f36ad-9d15-49f9-9660-6cf9746ba446",
    "orgid": "9b3f36ad-9d15-49f9-9660-6cf9746ba446",
    "referer": "HHG-1281b688-d639",
    "bezeichnung": "Englischkurs Klassen 3 und 4",
    "typ": "KURS",
    "bereich": "PFLICHT",
    "option": ["01"],
    "differenzierung": "G",
    "bildungsziele": ["RS"],
    "jahrgangsstufen": ["03", "04"],
    "faecher": [{
        "kennung": "EN"
    }],
    "referenzgruppen": [{
        "grupid": "e70663cb-65c4-4ad5-a9f1-7edbc3c36c4b",
        "rollen": ["Lern", "GMit"]
    },
    {
        "grupid": "11e84d3f-d78e-43b7-8c20-d26e43eb3761",
        "rollen": ["Lern"]
    }
    ]
}

```

```
    ],  
    "laufzeit": {  
        "von": "2022-08-01",  
        "bis": "2023-07-31"  
    },  
    "revision": "2"  
}
```

*Quellcode 121: Beispiel Response zu /gruppen/{gruppe.id} auf HTTP-PUT*

### Mögliche Fehlermeldungen

Zugrunde liegen die in 7.1.2 definierten Fehlermeldungen.

Darüber hinaus gibt es keine spezifischen Fehlermeldungen für diesen Endpunkt.

## DELETE

Dieser Schnittstellenendpunkt löscht die Gruppe zur angegebenen `{gruppe.id}`.

Ein DELETE ist nur mit HTTP-DELETE auf die API `/v1/gruppen/{gruppe.id}` zum Löschen einer Gruppe per `{gruppe.id}` zugelassen. Die Anfrage-Nutzdaten (Request Payload) beinhalten ein JSON-Objekt des Datentyps Gruppe. Das Datenmodell „Gruppe“ ist in 5.6 beschrieben.

Es ist erforderlich, dass für eine Löschanfrage einer Gruppe das Attribut `gruppe.revision` der zugrunde liegenden Gruppe mitgeschickt wird. Der SchulConneX-Server überprüft anhand des mitgeschickten Wertes des Attributs `gruppe.revision`, ob der Datensatz der Gruppe in der Zwischenzeit keine Änderung erfahren hat.

**Wichtig:** Das Löschen einer Gruppe löscht auch gleichzeitig alle zu der Gruppe gehörenden Gruppenzugehörigkeiten.

### Request

Nachfolgend die JSON-Struktur der Anfrage-Nutzdaten (Request Payload):

```
{
  "revision": <String>,
  // Revision der zugrunde liegenden Gruppe
} // Siehe Gruppe
```

*Quellcode 122: JSON-Struktur eines DELETE-Requests an `/v1/gruppen/{gruppe.id}`*

Beispiel Anfrage-Nutzdaten (Request Payload):

```
{
  "revision": "2"
}
```

*Quellcode 123: Beispiel Request zu `/v1/gruppen/{gruppe.id}` auf HTTP-DELETE*

### Response

Bei einer erfolgreichen Ausführung der Löschanfrage wird es keine Antwort-Nutzdaten (Response Payload) geben. Eine erfolgreiche Löschanforderung einer Gruppe per ID wird mit dem HTTP Status Code 204 („204 No Content“) quittiert.

### Mögliche Fehlermeldungen

Zugrunde liegen die in 7.1.2 definierten Fehlermeldungen.

Darüber hinaus gibt es keine spezifischen Fehlermeldungen für diesen Endpunkt.

### 10.3.3 [/gruppen/{gruppe.id}/gruppenzugehoerigkeiten] Erstellen und Lesen von zu einer Gruppe gehörenden Gruppenzugehörigkeiten

Die Schnittstelle /v1/gruppen/{gruppe.id}/gruppenzugehoerigkeiten bezieht sich auf das Anfordern der zu einer Gruppe gehörenden Gruppenzugehörigkeiten einer Gruppe per ID.

https://subdomain.domain.org/subpaths/v1/gruppen/{gruppe.id}/gruppenzugehoerigkeiten

Quellcode 124: URL-Struktur des Endpunkts gruppen/{gruppe.id}/gruppenzugehoerigkeiten

Der Pfad-Parameter {gruppe.id} bezieht sich auf die ID der Gruppe.

Bei Zugriffen auf den Endpunkt sind folgende Operationen und HTTP-Methoden erlaubt:

Operation	HTTP-Methode
CREATE	POST
READ	GET

Tabelle 36: Zugelassene Operationen auf dem Endpunkt /v1/gruppen/{gruppe.id}/gruppenzugehoerigkeiten

## CREATE

Dieser Schnittstellenendpunkt erstellt eine Gruppenzugehörigkeit zur angegebenen Gruppe per ID `{gruppe.id}`.

Ein CREATE ist nur mit HTTP-POST auf die API `/v1/gruppen/{gruppe.id}/gruppenzugehoerigkeiten` zum Erstellen von Gruppenzugehörigkeiten zu einer Gruppe zugelassen. Die Anfrage-Nutzdaten (Request Payload) beinhalten ein JSON-Objekt des Datentyps Gruppenzugehörigkeiten. Das Datenmodell „Gruppenzugehörigkeit“ ist in 5.7 beschrieben.

### Request

Die folgende Tabelle listet die Attribute einer Gruppenzugehörigkeit auf, welche von einem Quellsystem oder Dienst nicht verändert werden können.

Attribut	In den Anfrage Nutzdaten erforderlich?	Bemerkung
id	nein	ID der Gruppenzugehörigkeit wird vom SchulConneX-Server vergeben und ist eindeutig Dieses Attribut ist unveränderbar (immutable).
revision	nein	Revision der Gruppenzugehörigkeit wird vom SchulConneX-Server mit der Erstellung des Datensatzes sowie Aktualisierung generiert Dieser Wert kann nicht von Quellsystemen oder Diensten gesetzt werden.

*Tabelle 37: Liste von Attributen, welche durch ein Quellsystem oder Dienst über den Endpunkt `/v1/gruppen/{gruppe.id}/gruppenzugehoerigkeiten` auf HTTP-POST nicht gesetzt werden können.*

In Anbetracht der Attribute, die von einem Quellsystem oder Dienst bei der Erstellung einer Gruppenzugehörigkeit nicht gesetzt oder verändert werden können, ergibt sich folgende JSON-Struktur der Anfrage-Nutzdaten (Request Payload):

```
{
```



```

    "referrer": <String>, // Organisationseigene ID der
Gruppenzugehoerigkeit
    "ktid": <String>, // ID des Personenkontexts
    "rollen": [<Code>, <Code>, <Code>...], // Siehe Code-Liste
Gruppenrollen
    "von": <Datum>, // Anfangsdatum
    "bis": <Datum>, // Enddatum
}

```

*Quellcode 125: JSON-Struktur eines CREATE-Requests an /v1/gruppen/{gruppe.id}/gruppenzugehoerigkeiten*

### Beispiel Anfrage-Nutzdaten (Request Payload):

```

{
  "referrer": "78f85bd8-293d-11ed-a261-0242ac120002",
  "ktid": " e3bbcd03-7d91-4474-9de3-28a6bd162005",
  "rollen": ["LERN"],
  "von": "2022-08-01",
  "bis": "2023-07-31"
}

```

*Quellcode 126: Beispiel Request zu /v1/gruppen/{gruppe.id}/ gruppenzugehoerigkeiten auf HTTP-POST*

### Response

Bei einer erfolgreichen Anforderung zum Erstellen einer Gruppenzugehörigkeit zu einer Gruppe wird diese Anforderung mit einer Repräsentation der Gruppenzugehörigkeit in den Antwort-Nutzdaten (Response Payload) und dem HTTP Status Code 200 („200 OK“) quittiert.

### Nachfolgend die JSON-Struktur der Antwort-Nutzdaten (Response Payload):

```

{
  "id": <String>, // ID der Gruppenzugehoerigkeit
  "referrer": <String>, // Organisationseigene ID der
Gruppenzugehoerigkeit
  "ktid": <String>, // ID des Personenkontexts
  "rollen": [<Code>, <Code>, <Code>...] // Siehe Code-Liste
Gruppenrollen
  "von": <Datum>, // Anfangsdatum
  "bis": <Datum>, // Enddatum
  "revision": <String>, // Revisionsnummer
}

```

```
}
```

*Quellcode 127: JSON-Struktur einer CREATE-Response an /v1/gruppen/{gruppe.id}/gruppenzugehoerigkeiten*

### Beispiel Antwort-Nutzdaten (Response Payload):

```
{
  "id": " c9568463-a017-47a5-9a82-e274fbaa254a",
  "referrer": "78f85bd8-293d-11ed-a261-0242ac120002",
  "ktid": " e3bbcd03-7d91-4474-9de3-28a6bd162005",
  "rollen": ["LERN"],
  "von": "2022-08-01",
  "bis": "2023-07-31",
  "revision": "1"
}
```

*Quellcode 128: Beispiel Response zu /v1/gruppen/{gruppe.id}/gruppenzugehoerigkeiten auf HTTP-POST*

### Mögliche Fehlermeldungen

Zugrunde liegen die in 7.1.2. definierten Fehlermeldungen.

Darüber hinaus gibt es keine spezifischen Fehlermeldungen für diesen Endpunkt.

## READ

Dieser Schnittstellenendpunkt gibt die Gruppenzugehörigkeiten zur angegebenen `{gruppe.id}` zurück.

Ein READ ist nur mit HTTP-GET auf die API

`/v1/gruppen/{gruppe.id}/gruppenzugehoerigkeiten` zum Abfragen der Gruppenzugehörigkeiten per `gruppe.id` zugelassen. Die Antwort-Nutzdaten (Response Payload) beinhalten ein JSON-Objekt mit einem Array des Datentyps Gruppenzugehörigkeit. Das Datenmodell „Gruppenzugehörigkeit“ ist in 5.7 Gruppenzugehörigkeit beschrieben.

Die Schnittstelle `/v1/gruppen/{gruppe.id}/gruppenzugehoerigkeiten` ermöglicht das Verwenden von Filterparametern zur Präzisierung der Anfrage bei HTTP-GET.

Durch das Setzen von Filterparametern (query-parameter) wird mit der HTTP-GET-Anfrage eine Filteroperation mitgeschickt, welche die Ergebnismenge der Antwort-Nutzdaten (Response Payload) beeinflusst, beispielsweise mit den Filterparametern „`?referrer=123&rollen=LEHR`“.

Bei Filtern vom Typ “String” überprüft der SchulConneX-Server, ob der Wert des Filterparameters ohne Beachtung von Groß- und Kleinschreibung im Filter-Attribut beinhaltet ist (contains). Bei Filtern vom Typ “Code” überprüft der SchulConneX-Server, ob der Wert des Filterparameters ohne Beachtung von Groß- und Kleinschreibung gleich dem Filter-Attribut ist (equals). Werden mehrere Filter angegeben, so sind sie als mit 'und' verknüpft anzusehen.

Die mehrfache Nutzung des gleichen Filters in einer Anfrage ist nicht zulässig.

Folgende Tabelle listet die zur Verfügung stehenden Filter:

Filter	Typ	Beschreibung
referrer	String	Möglichkeit, die Ergebnisliste der Anfrage nach dem Fremdschlüssel “referrer” zu filtern
rollen	Code	Möglichkeit, die Ergebnisliste der Anfrage nach dem Attribut “gruppenzugehoerigkeit.rollen ” zu filtern Mehrere Rollen können durch Komma separiert werden.

Tabelle 38: Zugelassene Filter auf dem Endpunkt `/v1/gruppen/{gruppe.id}/gruppenzugehoerigkeiten` `/v1/gruppen`

## Request

Die Anfrage erfordert keine Nutzdaten (Request Payload).

## Response

Nachfolgend die JSON-Struktur der Antwort-Nutzdaten (Response Payload):

```
[
  {
    "id": <String>, // ID der Gruppenzugehoerigkeit
    "referrer": <String>,
    // ID des Gruppenzugehoerigkeit im Quellsystem.
    "rollen": [<Code>, <Code>, ...] // Siehe Code-Liste "Rolle"
    "von": <Datum>, // Beginn der Gruppenzugehoerigkeit
    "bis": <Datum>, // Ende der Gruppenzugehoerigkeit
    "revision": <String>
    // Revision des Datensatzes Gruppenzugehoerigkeit.
  },
]
```

*Quellcode 129: JSON-Struktur einer READ-Response an /v1/gruppen/{gruppe.id}/gruppenzugehoerigkeiten*

Beispiel Antwort-Nutzdaten (Response Payload):

```
[{
  "id": "2c09ca66-1897-11ed-861d-0242ac120002",
  "referrer": "78f85bd8-293d-11ed-a261-0242ac120002",
  "rollen": ["LERN"],
  "von": "2013-01-01",
  "bis": "2013-12-31",
  "revision": "1"
}]
```

*Quellcode 130: Beispiel Response zu /gruppen/{gruppe.id}/gruppenzugehoerigkeiten auf HTTP-GET*

## Mögliche Fehlermeldungen

Zugrunde liegen die in 7.1.2 definierten Fehlermeldungen.

Darüber hinaus gibt es keine spezifischen Fehlermeldungen für diesen Endpunkt.

#### 10.3.4 **[/gruppenzugehoerigkeiten]** Gruppenzugehörigkeiten der eigenen Organisation lesen

Die Schnittstelle /v1/gruppenzugehoerigkeiten bezieht sich auf das Anfordern und die Auflistung von Gruppenzugehörigkeiten in Bezug zu der eigenen Organisation.

<code>https://subdomain.domain.org/subpaths/v1/gruppenzugehoerigkeiten</code>
---

*Quellcode 131: URL-Struktur des Endpunkts gruppenzugehoerigkeiten*

Bei Zugriffen auf den Endpunkt ist folgende Operation und HTTP-Methode erlaubt:

Operation	HTTP-Methode
READ	GET

*Tabelle 39: Zugelassene Operation auf dem Endpunkt /v1/gruppenzugehoerigkeiten*

## READ

Dieser Schnittstellenendpunkt gibt alle Gruppenzugehörigkeiten zurück, auf die der Quellsystemanbieter zugreifen kann.

Ein READ kann nur mit einer HTTP-GET Anfrage erfolgen. Die Antwort-Nutzdaten (Response Payload) beinhalten ein Array von JSON-Objekten vom Datentyp Gruppendatensatz mit den jeweiligen Gruppenzugehörigkeiten und der zugeordneten Gruppe, sofern Einträge existieren.

Das Datenmodell „Gruppendatensatz“ ist in 5.5 beschrieben. Das Datenmodell „Gruppe“ ist in 5.6 beschrieben. Das Datenmodell „Gruppenzugehörigkeiten“ ist in 5.7 beschrieben.

Die Antwort-Nutzdaten umfassen standardmäßig für das Attribut {gruppe} die Gruppen-IDID über das Attribut {gruppe.id}.

Durch das Setzen von Filterparametern (query-parameter) wird mit der HTTP-GET-Anfrage eine Filteroperation mitgeschickt, welche die Ergebnismenge der Antwort-Nutzdaten (Response Payload) beeinflusst, beispielsweise mit dem Filterparameter „?rollen=LEHR“.

Bei Filtern vom Typ “String” überprüft der SchulConneX -Server, ob der Wert des Filterparameters ohne Beachtung von Groß- und Kleinschreibung im Filter-Attribut beinhaltet ist (contains). Bei Filtern vom Typ “Code” überprüft der SchulConneX-Server, ob der Wert des Filterparameters ohne Beachtung von Groß- und Kleinschreibung gleich dem Filter-Attribut ist (equals). Werden mehrere Filter angegeben, so sind sie als mit 'und' verknüpft anzusehen.

Filter	Typ	Beschreibung
referrer	String	Möglichkeit, die Ergebnisliste der Anfrage nach dem Fremdschlüssel “referrer” zu filtern
rollen	Code	Möglichkeit, die Ergebnisliste der Anfrage nach dem Attribut “gruppenzugehoerigkeit.rollen ” zu filtern

Tabelle 40: Zugelassene Filter auf dem Endpunkt /v1/gruppenzugehoerigkeiten

## Request

Die Anfrage erfordert keine Nutzdaten (Request Payload).

## Response

Nachfolgend die JSON-Struktur der Antwort-Nutzdaten (Response Payload):

[
---

```

{
  "gruppe": {
    "id": <String> // ID der Gruppe.
  },
  "gruppenzugehoerigkeiten": [
    {
      "id": <String>, // ID der Gruppenzugehoerigkeit
      "referrer": <String>,
        // ID des Gruppenzugehoerigkeit im Quellsystem.
      "ktid": <String>, // ID des Personenkontexts
      "rollen": [<Code>, <Code>, <Code>...] // Siehe Code-Liste
Gruppenrollen
      "von": <Datum>, // Anfangsdatum
      "bis": <Datum>, // Enddatum
      "revision": <String>, // Revisionsnummer
    }
  ]
}
]

```

*Quellcode 132: JSON-Struktur einer READ-Response an /v1/gruppenzugehoerigkeiten*

#### Beispiel Antwort-Nutzdaten (Response Payload):

```

[ {
  " gruppe ": {
    "id": "c7de2c10-26a6-4d12-8152-f9e8db497625"
  },
  "gruppenzugehoerigkeiten": [ {
    "id": " c9568463-a017-47a5-9a82-e274fbaa254a",
    "referrer": "78f85bd8-293d-11ed-a261-0242ac120002",
    "ktid": " e3bbcd03-7d91-4474-9de3-28a6bd162005",
    "rollen": ["LERN"],
    "von": "2022-08-01",
    "bis": "2023-07-31",
    "revision": "1"
  } ]
} ]

```

*Quellcode 133: Beispiel Response zu /gruppenzugehoerigkeiten auf HTTP-GET*

### Mögliche Fehlermeldungen

Zugrunde liegen die in 7.1.2. definierten Fehlermeldungen.

Darüber hinaus gibt es keine spezifischen Fehlermeldungen für diesen Endpunkt.



### 10.3.5 [/gruppenzugehoerigkeiten/{gruppenzugehoerigkeit.id}]

Lesen, ändern und löschen einer individuellen Gruppenzugehörigkeit

Die Schnittstelle /v1/gruppenzugehoerigkeiten/{gruppenzugehoerigkeit.id} bezieht sich auf das Anfordern einer Gruppenzugehörigkeit per ID.

```
https://subdomain.domain.org/subpaths/v1/gruppenzugehoerigkeiten/{gruppenzugehoerigkeit.id}
```

*Quellcode 134: URL-Struktur des Endpunkts gruppenzugehoerigkeiten/{gruppenzugehoerigkeit.id}*

Der Pfad-Parameter {gruppenzugehoerigkeit.id} bezieht sich auf die ID der Gruppenzugehörigkeit.

Bei Zugriffen auf den Endpunkt sind folgende Operationen und HTTP-Methoden erlaubt:

Operation	HTTP-Methode
READ	GET
UPDATE	PUT
DELETE	DELETE

*Tabelle 41: Zugelassene Operationen auf dem Endpunkt /v1/gruppenzugehoerigkeiten/{gruppenzugehoerigkeit.id}*

## READ

Dieser Schnittstellenendpunkt gibt die Gruppenzugehörigkeit zur angegebenen {gruppenzugehoerigkeit.id} zurück.

Ein READ ist nur mit HTTP-GET auf die API

/v1/gruppenzugehoerigkeiten/{gruppenzugehoerigkeit.id} zum Abfragen einer Gruppenzugehörigkeit per ID zugelassen. Die Antwort-Nutzdaten (Response Payload) beinhalten ein JSON-Objekt des Datentyps Gruppendatensatz. Das Datenmodell „Gruppendatensatz“ ist in 5.8 Gruppendatensatz beschrieben.

Aus Konsistenzgründen mit dem Datenmodell Gruppendatensatz wird das Attribut "gruppenzugehoerigkeiten" als Array geliefert, auch wenn aufgrund des Zugriffs über die eindeutige ID {gruppenzugehoerigkeit.id} immer nur genau ein Element geliefert wird.

## Request

Die Anfrage erfordert keine Nutzdaten (Request Payload).

## Response

Nachfolgend die JSON-Struktur der Antwort-Nutzdaten (Response Payload):

```
{
  "gruppe": {
    "id": <String>, // ID der Gruppe
    "mandant": <String>, // ID des Mandanten
    "orgid": <String>, // ID der Organisation
    "referrer": <String>, // Organisationseigene ID der Gruppe
    "bezeichnung": <String>, //Bezeichnung der Gruppe
    "thema": <String>, //Optionales Thema der Gruppe
    "beschreibung": <String>, //Optionale Beschreibung der Gruppe
    "typ": <Code>, // Siehe Code-Liste Gruppentyp
    "bereich": <Code>, // Siehe Code-Liste Gruppen
    "optionen": [<Code>, <Code>, <Code>...], // Siehe Code-Liste
    Gruppenoption
    "differenzierung": <Code>, // Siehe Code-Liste
    Gruppendifferenzierung
    "bildungsziele": [<Code>, <Code>, <Code>...] // Siehe Code-Liste
    "jahrgangsstufen": [<Code>, <Code>, <Code>...] // Siehe Code-Liste
    Jahrgangsstufe
    "faecher": [<faecher>, <faecher>, <faecher>...] // Objekte mit
    Faecher Attributen, siehe Beschreibung "Gruppe"
```

```

    "laufzeit": { // von/vonlernperiode und bis/bislernperiode sind
jeweils alternativ
        "von": <Datum>, // Anfangsdatum
        "vonlernperiode": <Code>, // Referenz auf Codeliste Lernperiode
        "bis": <Datum>, // Enddatum
        "bislernperiode": <Code>, // Referenz auf Codeliste Lernperiode
    }
}
"gruppenzugehoerigkeiten": [
{
    "id": <String>, // ID der Gruppenzugehoerigkeit
    "referrer": <String>,
        // ID des Gruppenzugehoerigkeit im Quellsystem.
    "ktid": <String>, // ID des Personenkontexts
    "rollen": [<Code>, <Code>, <Code>...] // Siehe Code-Liste
Gruppenrollen
    "von": <Datum>, // Anfangsdatum
    "bis": <Datum>, // Enddatum
    "revision": <String>, // Revisionsnummer
}
]
}

```

*Quellcode 135: JSON-Struktur einer READ-Response an /v1/gruppenzugehoerigkeiten/{gruppenzugehoerigkeit.id}*

#### Beispiel Antwort-Nutzdaten (Response Payload):

```

{
    "gruppe": {
        "id": "c7de2c10-26a6-4d12-8152-f9e8db497625",
        "orgid": "9b3f36ad-9d15-49f9-9660-6cf9746ba446",
        "referer": "HHG-1281b688-d639",
        "bezeichnung": "Englischkurs Klasse 3b",
        "typ": "KURS",
        "bereich": "PFLICHT",
        "optionen": ["01"],
        "differenzierung": "G",
        "bildungsziele": ["RS"],
        "jahrgangsstufen": ["03"],
    }
}

```

```

        "faecher": [{
            "kennung": "EN"
        }],
        "laufzeit": {
            "von": "2022-08-01",
            "bis": "2023-07-31"
        },
        "revision": "1"
    },
    "gruppenzugehoerigkeiten": [{
        "id": "c9568463-a017-47a5-9a82-e274fbaa254a",
        "referrer": "78f85bd8-293d-11ed-a261-0242ac120002",
        "ktid": " e3bbcd03-7d91-4474-9de3-28a6bd162005",
        "rollen": ["LERN"],
        "von": "2022-08-01",
        "bis": "2023-07-31",
        "revision": "1"
    }]
}

```

*Quellcode 136: Beispiel Response zu /gruppenzugehoerigkeiten/{gruppenzugehoerigkeit.id} auf HTTP-GET*

### Mögliche Fehlermeldungen

Zugrunde liegen die in 7.1.2. definierten Fehlermeldungen.

Darüber hinaus gibt es keine spezifischen Fehlermeldungen für diesen Endpunkt.

## UPDATE

Dieser Schnittstellenendpunkt aktualisiert die Gruppenzugehörigkeit zur angegebenen `{gruppenzugehoerigkeit.id}`.

Ein UPDATE ist nur mit HTTP-PUT auf die API `/v1/gruppenzugehoerigkeiten/{gruppenzugehoerigkeit.id}` zum Aktualisieren einer Gruppenzugehörigkeit per ID zugelassen. Die Anfrage-Nutzdaten (Request Payload) beinhalten ein JSON-Objekt des Datentyps Gruppenzugehörigkeit. Das Datenmodell „Gruppenzugehörigkeit“ ist in 5.7 beschrieben.

Es ist erforderlich, dass zur Aktualisierungsanfrage in der bearbeiteten Gruppenzugehörigkeit das Attribut „revision“ der zugrunde liegenden Gruppenzugehörigkeit mitgeschickt wird. Der SchulConneX-Server überprüft anhand des mitgeschickten Wertes des Attributs „revision“, ob der Datensatz der Gruppenzugehörigkeit in der Zwischenzeit keine Änderung erfahren hat. Der Wert des Attributs kann von einem Quellsystem oder Dienst nicht verändert werden; dieser wird mit der Erstellung (Create) des Datensatzes Gruppenzugehörigkeit und jeder erfolgreichen Änderung (Update) vom SchulConneX-Server neu generiert.

Die folgende Tabelle listet die Attribute einer Gruppenzugehörigkeit auf, welche von einem Quellsystem oder Dienst nicht verändert werden können.

Attribut	Bemerkung
id	ID der Gruppenzugehörigkeit wird vom SchulConneX-Server vergeben und ist eindeutig Dieses Attribut ist unveränderbar (immutable).
revision	Revision der Gruppenzugehörigkeit wird vom SchulConneX-Server mit der Erstellung des Datensatzes sowie Aktualisierung generiert Dieser Wert kann nicht von Quellsystemen oder Diensten gesetzt werden.

*Tabelle 42: Liste von Attributen, welche durch ein Quellsystem oder Dienst über den Endpunkt `/v1/gruppenzugehoerigkeit/{gruppenzugehoerigkeit.id}` auf HTTP-PUT nicht verändert werden können.*

Die ID kann im Request ausgelassen werden. Ist das Attribut im Request dennoch vorhanden, wird es validiert und führt gegebenenfalls zu einer Fehlermeldung.

## Request

In Anbetracht der Attribute, die von einem Quellsystem oder Dienst bei der Aktualisierung einer Gruppenzugehörigkeit nicht gesetzt oder verändert werden können, ergibt sich folgende JSON-Struktur der Anfrage-Nutzdaten (Request Payload):

```
{
  "id": <String>, // ID der Gruppenzugehoerigkeit
  "referrer": <String>,
               // ID des Gruppenzugehoerigkeit im Quellsystem.
  "ktid": <String>, // ID des Personenkontexts
  "rollen": [<Code>, <Code>, <Code>...] // Siehe Code-Liste
Gruppenrollen
  "von": <Datum>, // Anfangsdatum
  "bis": <Datum>, // Enddatum
  "revision": <String>, // Revisionsnummer
}
```

*Quellcode 137: JSON-Struktur eines UPDATE-Requests an /v1/gruppenzugehoerigkeiten/{gruppenzugehoerigkeit.id}*

Anmerkung: Bei dieser Aktualisierungsanfrage handelt es sich um eine HTTP-PUT-Operation, bei der die Anfrage-Nutzlast mit den gegebenen Attributen der Aktualisierung zugrunde gelegt wird (mitgeschickte Attribute werden aktualisiert, nicht mitgeschickte Attribute werden gelöscht); es ist keine HTTP-PATCH-Operation, bei der lediglich die zu aktualisierenden Attribute in der Anfrage-Nutzlast mitgeschickt werden.

## Response

Bei einer erfolgreichen Anforderung zum Aktualisieren einer Gruppenzugehörigkeit wird diese Anforderung mit einer Repräsentation der Gruppenzugehörigkeit in den Antwort-Nutzdaten (Response Payload) und dem HTTP Status Code 200 („200 OK“) quittiert.

Nachfolgend die JSON-Struktur der Antwort-Nutzdaten (Response Payload):

```
{
  "id": <String>, // ID der Gruppenzugehoerigkeit
  "referrer": <String>,
               // ID des Gruppenzugehoerigkeit im Quellsystem.
  "ktid": <String>, // ID des Personenkontexts
  "rollen": [<Code>, <Code>, <Code>...] // Siehe Code-Liste
Gruppenrollen
  "von": <Datum>, // Anfangsdatum
  "bis": <Datum>, // Enddatum
  "revision": <String>, // Revisionsnummer
}
```

```
}
```

*Quellcode 138: JSON-Struktur einer UPDATE-Response an /v1/gruppenzugehoerigkeiten/{gruppenzugehoerigkeit.id}*

#### Beispiel Antwort-Nutzdaten (Response Payload):

```
{
  "id": "c9568463-a017-47a5-9a82-e274fbaa254a",
  "referrer": "78f85bd8-293d-11ed-a261-0242ac120002",
  "ktid": " e3bbcd03-7d91-4474-9de3-28a6bd162005",
  "rollen": ["LERN"],
  "von": "2022-09-01",
  "bis": "2023-07-31",
  "revision": "2"
}
```

*Quellcode 139: Beispiel Response zu /gruppenzugehoerigkeiten/{gruppenzugehoerigkeit.id } auf HTTP-PUT*

#### Mögliche Fehlermeldungen

Zugrunde liegen die in 7.1.2 definierten Fehlermeldungen.

Darüber hinaus gibt es keine spezifischen Fehlermeldungen für diesen Endpunkt.

## DELETE

Dieser Schnittstellenendpunkt löscht die Gruppenzugehörigkeit zur angegebenen `{gruppenzugehoerigkeit.id}`.

Ein DELETE ist nur mit HTTP-DELETE auf die API `/v1/gruppenzugehoerigkeit/{gruppenzugehoerigkeit.id}` zum Löschen einer Gruppe per `{gruppe.id}` zugelassen. Die Anfrage-Nutzdaten (Request Payload) beinhalten ein JSON-Objekt des Datentyps Gruppenzugehörigkeit. Das Datenmodell „Gruppenzugehörigkeit“ ist in 5.7 beschrieben.

Es ist erforderlich, dass für eine Löschanfrage einer Gruppenzugehörigkeit das Attribut „revision“ der zugrunde liegenden Gruppenzugehörigkeit mitgeschickt wird. Der SchulConneX-Server überprüft anhand des mitgeschickten Wertes des Attributs „revision“, ob der Datensatz des Personenkontext in der Zwischenzeit keine Änderung erfahren hat.

### Request

Nachfolgend die JSON-Struktur der Anfrage-Nutzdaten (Request Payload):

```
{
  "revision": <String>,
  // Revision der zugrunde liegenden Gruppenzugehoerigkeit
} // Siehe Gruppenzugehoerigkeit
```

*Quellcode 140: JSON-Struktur eines DELETE-Requests an `/v1/gruppenzugehoerigkeiten/{gruppenzugehoerigkeit.id}`*

Beispiel Anfrage-Nutzdaten (Request Payload):

```
{
  "revision": "2"
}
```

*Quellcode 141: Beispiel Request zu `/v1/gruppenzugehoerigkeiten/{gruppenzugehoerigkeit.id}` auf HTTP-DELETE*

### Response

Bei einer erfolgreichen Ausführung der Löschanfrage wird es keine Antwort-Nutzdaten (Response Payload) geben. Eine erfolgreiche Löschanforderung einer Gruppenzugehörigkeit per ID wird mit dem HTTP Status Code 204 („204 No Content“) quittiert.

### Mögliche Fehlermeldungen

Zugrunde liegen die in 7.1.2 definierten Fehlermeldungen.

Darüber hinaus gibt es keine spezifischen Fehlermeldungen für diesen Endpunkt.



## 11 Allgemeine API-Endpunkte

### 11.1 Schnittstellen für Codelisten

#### 11.1.1 [/codelisten] Auflistung verfügbarer Codelisten

Die Schnittstelle /v1/codelisten bezieht sich auf die verfügbaren Codelisten, welche Verwendung in den Schnittstellen finden.

`https://subdomain.domain.org/subpaths/v1/codelisten`

*Quellcode 142: URL-Struktur des Endpunkts codelisten*

Bei Zugriffen auf den Endpunkt ist folgende Operation und HTTP-Methode erlaubt:

Operation	HTTP-Methode
READ	GET

*Tabelle 43: Zugelassene Operation auf dem Endpunkt /v1/codelisten*

*READ*

#### Request

Die Anfrage erfordert keine Nutzdaten (Request Payload).

#### Response

Die Antwort beinhaltet (Response Payload) ein Array von Strings.

Nachfolgend die JSON-Struktur der Antwort-Nutzdaten (Response Payload):

```
[
  "geschlecht",
  "personenstatus",
  "rolle",
  "vertrauensstufe",
  "traegerschaft"
  ...
]
```

Quellcode 143: JSON-Struktur eines READ-Requests an /v1/codelisten

### 11.1.2 [/codelisten/{codeliste}] Auflistung des Inhalts einer Codeliste

Die Schnittstelle /v1/codelisten/{codeliste} bezieht sich auf den Inhalt einzelner, verfügbarer Codelisten.

```
https://subdomain.domain.org/subpaths/v1/codelisten/{codeliste}
```

Quellcode 144: URL-Struktur des Endpunkts codelisten/{codeliste}

Bei Zugriffen auf den Endpunkt ist folgende Operation und HTTP-Methode erlaubt:

Operation	HTTP-Methode
READ	GET

Tabelle 44: Zugelassene Operation auf dem Endpunkt /v1/codelisten/{codeliste}

#### READ

#### Request

Die Anfrage erfordert keine Nutzdaten (Request Payload).

#### Response

Die Antwort (Response Payload) beinhaltet ein Array von Objekten, welche die Codes der angefragten Liste sowie Beschreibungen zu den jeweiligen Codes beinhalten.

Nachfolgend die JSON-Struktur der Antwort-Nutzdaten (Response Payload):

```
{
  "geschlecht": [
    {"code": "m", "beschreibung": "männlich"},
    {"code": "w", "beschreibung": "weiblich"},
    {"code": "d", "beschreibung": "divers"},
    {"code": "x", "beschreibung": "Keine Angabe"}
  ]
}
```

Quellcode 145: JSON-Struktur eines READ-Requests an /v1/codelisten/geschlecht

## 11.2 Schnittstellen für die Versionsverwaltung

### 11.2.1 [/versionen] Auflistung verfügbarer Codelisten

Die Schnittstelle /v1/versionen bezieht sich auf die vom Server angebotenen Versionen der API.

```
https://subdomain.domain.org/subpaths/v1/versionen
```

Quellcode 146: URL-Struktur des Endpunkts *versionen*

Bei Zugriffen auf den Endpunkt ist folgende Operation und HTTP-Methode erlaubt:

Operation	HTTP-Methode
READ	GET

Tabelle 45: Zugelassene Operation auf dem Endpunkt */v1/versionen*

*READ*

### Request

Die Anfrage erfordert keine Nutzdaten (Request Payload).

### Response

Die Antwort (Response Payload) beinhaltet ein Array von Objekten, welche die vom Server bereitgestellten Versionen der API auflistet.

Die Versionsnummer besteht aus drei, durch einen Punkt getrennten Zahlen. Dabei handelt es sich um die Hauptversions-Nummer, die Unterversions-Nummer und die Patch-Nummer.

Die Hauptversions-Nummer ändert sich, wenn eine Version der API nicht mehr mit der vorherigen Version kompatibel ist, also Programme, welche die Antworten einer älteren API erwarten, sich voraussichtlich fehlerhaft verhalten.

Die Unterversions-Nummer ändert sich, wenn sich die API geändert hat, die Änderungen aber voraussichtlich keine Auswirkung auf Programme hat, welche diese API nutzen. Beispiel für solche Änderungen ist die Hinzufügung neuer Optionen oder APIs, welche das Verhalten bestehender APIs nicht beeinflussen.

Die Patch-Nummer ändert sich, wenn sich an der Definition der API nichts geändert hat, was einen Einfluss auf Programme hat, jedoch Modifikationen am Server vorgenommen wurden, beispielsweise aufgrund von Fehlerbehebungen oder inhaltlichen Erweiterungen bei gleicher API.

Nachfolgend die JSON-Struktur der Antwort-Nutzdaten (Response Payload):

```
{
  "versionen": [
    {"version": "1.0021.0211",
     "path": "https://subdomain.domain.org/subpaths/v1/"},
    {"version": "2.0043.0211",
     "path": "https://subdomain.domain.org/subpaths/v2/"},
  ]
}
```

```
{
  "version": "2.0124.0432",
  "path": "https://subdomain.domain.org/subpaths/v2test/",
}
]
```

*Quellcode 147: JSON-Struktur eines READ-Requests an /v1/versionen*

Hierbei gibt die erste gelistete Version immer die Version der API an, welche gerade aufgerufen wurde. Werden von einem Server weitere Versionen der API unter anderen URL-Pfaden angeboten, so können diese zur aktuellen Version mitgeliefert werden. Dabei wird immer der URL-Pfad bis einschließlich der Versionsnummer und einem abschließenden Schrägstrich geliefert.

## 12 Codelisten

Codelisten legen mögliche Werte für bestimmte Attribute fest, beispielsweise für „Geschlecht“ oder „Organisationstyp“. Nachfolgend befinden sich alle Codelisten, welche in der Spezifikation referenziert werden.

Codes aus Codelisten werden generell als Strings zurückgegeben, d. h. die JSON-Form für das weibliche Geschlecht ist als *„geschlecht“:“w“* und nicht *„geschlecht“:w* zu erwarten.

Alle Codes sind als nicht abhängig von Groß-/Kleinschreibung zu interpretieren (sind nicht Case-sensitiv). Die Organisationsarten „SCHULE“, „schule“, „Schule“ oder auch „SchuLe“ sind als identisch anzunehmen.

### 12.1 Personenstatus

Code	Bezeichnung
<b>AKTIV</b>	aktiv

*Codeliste 2: Personenstatus*

### 12.2 Geschlecht

Code	Bezeichnung
<b>m</b>	männlich
<b>w</b>	weiblich
<b>d</b>	divers
<b>x</b>	keine Angabe

*Codeliste 1: Geschlecht*

### 12.3 Rolle

Code	Bezeichnung
<b>LERN</b>	Lernende/r
<b>LEHR</b>	Lehrende/r
<b>SORGBER</b>	Sorgeberechtigte/r
<b>EXTERN</b>	externe Person
<b>ORGADMIN</b>	Organisationsadministrator
<b>LEIT</b>	Organisationsleitung
<b>SYSADMIN</b>	Systemadministrator

Codeliste 3: Rolle

### 12.4 Vertrauensstufe

Code	Bezeichnung	Bemerkung
<b>KEIN</b>	keine	Der Schlüsselbesitzer ist nicht vertrauenswürdig. Diese Vertrauensstufe wird meist für Test- oder Beispiel-Nutzer vergeben.
<b>UNBE</b>	unbekannt	Der Schlüsselbesitzer ist unbekannt. Diese Vertrauensstufe wird beispielsweise für selbstregistrierte Nutzer vergeben.
<b>TEIL</b>	vertraut	Dem Schlüsselbesitzer wird vertraut. Hierbei handelt es sich meist um Nutzer, die von Nutzern mit höherer Stufe angelegt wurden und diesen bekannt sind, sich jedoch nicht einer formalen Identifizierung unterzogen haben.
<b>VOLL</b>	verifiziert	Der Schlüsselbesitzer ist eindeutig und formal identifiziert (beispielsweise durch Vorlage des Personalausweises).

Codeliste 4: Vertrauensstufe

## 12.5 Organisationstyp

Code	Bezeichnung
<b>SCHULE</b>	Schule
<b>ANBIETER</b>	Anbieter
<b>SONSTIGE</b>	sonstige Organisation / Einrichtungen

*Codeliste 5: Organisationstyp*

## 12.6 Jahrgangsstufe

Code	Bezeichnung
<b>01</b>	Jahrgangsstufe 1
<b>02</b>	Jahrgangsstufe 2
<b>03</b>	Jahrgangsstufe 3
<b>04</b>	Jahrgangsstufe 4
<b>05</b>	Jahrgangsstufe 5
<b>06</b>	Jahrgangsstufe 6
<b>07</b>	Jahrgangsstufe 7
<b>08</b>	Jahrgangsstufe 8
<b>09</b>	Jahrgangsstufe 9
<b>10</b>	Jahrgangsstufe 10
<b>11</b>	Jahrgangsstufe 11
<b>12</b>	Jahrgangsstufe 12
<b>13</b>	Jahrgangsstufe 13

*Codeliste 6: Jahrgangsstufe*

## 12.7 Boolean

Die Codeliste „Boolean“ regelt das Mapping der deutschen Wörter „ja“ und „nein“ auf die englischen Begriffe „true“ und „false“.

Code	Bemerkung
JA	Mapping auf „true“.
NEIN	Mapping auf „false“.

Codeliste 7: Boolean

## 12.8 Trägerschaft

Code	Bezeichnung
01	Bund
02	Land
03	Kommune
04	Privat
05	Kirchlich
06	Sonstige

Codeliste 8: Trägerschaft

## 12.9 Lokalisierung

Basierend auf RFC 5646<sup>1</sup>. Beispiele finden sich in nachfolgender Tabelle.

Code	Bezeichnung
de	deutsch
de-XX	deutsch, Anrede du, einfache Sprache
en-GB	englisch (Vereinigtes Königreich)

Codeliste 9: Lokalisierung



## 12.10 Gruppenbereich

Hinweis: Diese Codeliste ist spezifisch für das Bundesland Niedersachsen. Bei Verwendung dieser API in anderen Bundesländern kann gegebenenfalls eine andere Codeliste Verwendung finden.

Code	Bezeichnung
<b>Pflicht</b>	Pflichtunterricht
<b>Wahl</b>	Wahlunterricht
<b>Wahlpflicht</b>	Wahlpflichtunterricht

*Codeliste 10: Gruppenbereich*

## 12.11 Gruppendifferenzierung

Hinweis: Diese Codeliste ist spezifisch für das Bundesland Niedersachsen. Bei Verwendung dieser API in anderen Bundesländern kann gegebenenfalls eine andere Codeliste Verwendung finden.

Code	Bezeichnung
<b>G</b>	G-Kurs
<b>E</b>	E-Kurs
<b>Z</b>	Z-Kurs
<b>gA</b>	grundlegendes Anforderungsniveau
<b>eA</b>	erhöhtes Anforderungsniveau

*Codeliste 11: Gruppendifferenzierung*

## 12.12 Gruppenoption

Code	Bezeichnung
<b>01</b>	bilingual
<b>02</b>	herkunftssprachlich

*Codeliste 12: Gruppenoption*

### 12.13 Gruppentyp

Code	Bezeichnung
Klasse	Schulklasse
Kurs	Kurs/Unterricht
Sonstig	Sonstige Gruppe

Codeliste 13: Gruppentyp

### 12.14 Gruppenrolle

Code	Bezeichnung
Lern	Schülerin/Schüler
Lehr	Lehrkraft
KlLeit	Klassenleitung
Foerd	Förderlehrkraft
SchB	Schulbegleitung
GMit	Gruppenmitglied
GLEit	Gruppenleitung

Codeliste 14: Gruppenrolle

### 12.15 Lernperiode

Code	Bezeichnung	Beginn	Ende	Typ
2022	Schuljahr 2022/23	2022-08-01	2023-07-31	SJ
2022-1	1. Halbj. 22/23	2022-08-01	2023-01-31	HJ
2022-2	2. Halbj. 22/23	2023-02-01	2023-07-31	HJ
2023	Schuljahr 2023/24	2023-08-01	2024-07-31	SJ
2023-1	1. Halbj. 23/24	2023-08-01	2024-01-31	HJ
2023-2	2. Halbj. 23/24	2024-02-01	2024-07-31	HJ
2024	Schuljahr 2024/25	2024-08-01	2025-07-31	SJ
2024-1	1. Halbj. 24/25	2024-08-01	2025-01-31	HJ

<b>2024-2</b>	2. Halbj. 24/25	2025-02-01	2025-07-31	HJ
<b>2025</b>	Schuljahr 2025/26	2025-08-01	2026-07-31	SJ
<b>2025-1</b>	1. Halbj. 25/26	2025-08-01	2026-01-31	HJ
<b>2025-2</b>	2. Halbj. 25/26	2026-02-01	2026-07-31	HJ
<b>2026</b>	Schuljahr 2026/27	2026-08-01	2027-07-31	SJ
<b>2026-1</b>	1. Halbj. 26/27	2026-08-01	2027-01-31	HJ
<b>2026-2</b>	2. Halbj. 26/27	2027-02-01	2027-07-31	HJ
<b>2027</b>	Schuljahr 2027/28	2027-08-01	2028-07-31	SJ
<b>2027-1</b>	1. Halbj. 27/28	2027-08-01	2028-01-31	HJ
<b>2027-2</b>	2. Halbj. 27/28	2028-02-01	2028-07-31	HJ

Codeliste 15: Lernperiode

## 12.16 Lernperiodentyp

Code	Bezeichnung
<b>SJ</b>	Schuljahr
<b>HJ</b>	Schulhalbjahr

Codeliste 16: Lernperiodentyp

## 12.17 Fächerkanon

Hinweis: Diese Codeliste ist spezifisch für das Bundesland Niedersachsen. Bei Verwendung dieser API in anderen Bundesländern kann gegebenenfalls eine andere Codeliste Verwendung finden.

Code	Bezeichnung
<b>BI</b>	Biologie
<b>CH</b>	Chemie
<b>CI</b>	Chinesisch
<b>DE</b>	Deutsch
<b>DS</b>	Darstellendes Spiel
<b>EK</b>	Erdkunde

<b>EN</b>	Englisch
<b>FR</b>	Französisch
<b>GR</b>	Griechisch
<b>NL</b>	Niederländisch
<b>IT</b>	Italienisch
<b>SN</b>	Spanisch
<b>KU</b>	Kunst
<b>LA</b>	Latein
<b>RS</b>	Russisch
<b>GE</b>	Geschichte
<b>PO</b>	Politik
<b>PW</b>	Politik/Wirtschaft
<b>RE</b>	Evangelische Religion
<b>RI</b>	Islamische Religion
<b>RK</b>	Katholische Religion
<b>SP</b>	Sport
<b>SU</b>	Sachunterricht
<b>TE</b>	Technik
<b>TG</b>	Textiles Gestalten
<b>WE</b>	Gestaltendes Werken
<b>WN</b>	Werte und Normen
<b>WS</b>	Wirtschaft
<b>DA</b>	Deutsch als Zweitsprache
<b>MA</b>	Mathematik

<b>HW</b>	Hauswirtschaft
<b>MU</b>	Musik
<b>PA</b>	Pädagogik
<b>PH</b>	Physik
<b>IF</b>	Informatik
<b>AW</b>	Arbeit-Wirtschaft-Technik
<b>GL</b>	Gesellschaftslehre
<b>PWI</b>	Profil Wirtschaft
<b>PTE</b>	Profil Technik
<b>PGUS</b>	Profil Gesundheit und Soziales
<b>NAT</b>	Naturwissenschaften

Codeliste 17: Fächerkanon

### 12.18 Bildungsziel

Diese Codeliste ist derzeit spezifisch für Niedersachsen. Bei Verwendung dieser API in anderen Bundesländern können die Codelisten aber auch voneinander abweichen.

<b>Code</b>	<b>Bezeichnung</b>
<b>GS</b>	Grundschule
<b>HS</b>	Hauptschule
<b>RS</b>	Realschule
<b>GY-SEK-I</b>	Gymnasium Sekundarstufe I
<b>GY-SEK-II</b>	Gymnasium Sekundarstufe II

Codeliste 18: Fächerkanon

### 12.19 Erreichbarkeitstyp

Erreichbarkeitstyp ist eine Liste von Kommunikationskanälen, über die eine Person oder Organisation erreicht werden kann.

<b>Code</b>	<b>Bezeichnung</b>	<b>Bemerkung</b>
-------------	--------------------	------------------

<b>E-Mail</b>	E-Mail	E-Mail-Adressen müssen RFC 5322 (Internet Message Format) erfüllen, um akzeptiert zu werden, anderenfalls wird ein 400/06 Fehler geliefert.
---------------	--------	---

Codeliste 19: Erreichbarkeitstyp

## 12.20 Beziehungen

Beziehungen ist eine Liste von Beziehungen, welche zwischen zwei Personen innerhalb von Personenkontexten bestehen können. Häufig wird hierüber eine Beziehung zwischen Erziehungsberechtigten und Lernenden hergestellt, aber auch Beziehungen wie Tutoren oder Tutorinnen zu Lernenden innerhalb einer Schule können darüber abgebildet werden.

Code	Bezeichnung	Bemerkung
<b>SorgBer</b>	Sorgeberechtigter oder Sorgeberechtigte	Hat eine Person A im Personenkontext diese ‚hat_als‘ Beziehung zu einem Personenkontext der Person B, so hat A als Sorgeberechtigten oder Sorgeberechtigte die Person B.
<b>SchB</b>	Schulbegleiter oder Schulbegleiterin	Hat eine Person A im Personenkontext diese ‚hat_als‘ Beziehung zu einem Personenkontext der Person B, so hat A als Schulbegleiter oder Schulbegleiterin die Person B.

Codeliste 20: Beziehungen

## 12.21 Organisationsbeziehungen

Organisationsbeziehungen ist eine Codeliste von Beziehungen, welche zwischen zwei Organisationen bestehen können.

Code	Bezeichnung	Bemerkung
<b>SchTrae</b>	Schulträger	Hat eine Organisation A diese ‚hat_als‘ Beziehung zu einer Organisation B, so hat A als Schulträger die Organisation B, damit ist B also Schulträger von A.
<b>SchBeh</b>	Schulbehörde	Hat eine Organisation A diese ‚hat_als‘ Beziehung zu einer Organisation B, so hat A als Schulbehörde die Organisation B, damit ist B also Schulbehörde von A.

Codeliste 21: Organisationsbeziehungen



## 13 Glossar

Begriff	Erklärung
Benutzerkonto (Identität)	Natürliche Person, die ein Anwendungssystem benutzt und sich dabei mittels ihrer Identität (auch persönliches Benutzerkonto/User Account) authentifiziert (in der Regel durch eine Benutzerkennung und ein Passwort). Personen können über mehrere Identitäten verfügen, eine Identität ist aber immer eindeutig einer Person zugeordnet.
Dienst	Eine Anwendung eines Dienstansbieters, welche die Schnittstelle aufruft.
Dienstanbieter	Im Sinne der Schnittstelle Anbietende von digitalen Diensten aus dem Bereich der Bildung (z. B. Bildungsmedien, Lernplattformen, Lernstandermittlungen).
Freigaben	Organisationen können individuelle Datensätze zur Einsicht für andere Organisationen freigeben. Alle Freigaben sind derzeit Sichtfreigaben.
HTTP	Das Hypertext Transfer Protocol ist das Übertragungsprotokoll für Daten. In den meisten Fällen wird heutzutage die sicherere Version HTTPS eingesetzt.
HTTPS	HTTPS steht für Hypertext Transfer Protocol Secure und bietet eine verschlüsselte Kommunikation zwischen Client und Server. Zusätzlich wird der Webserver durch ein Zertifikat authentifiziert, welches die Vertrauenswürdigkeit der Domäne bescheinigt. Als Verschlüsselungsprotokoll wird TLS genutzt.
JSON	JSON ist die Abkürzung für JavaScript Object Notation und wird als kompaktes Datenformat strukturierter Daten eingesetzt. Ein JSON besteht aus verschiedenen Elementen. Beispielsweise ist JSON-Objekt eine ungeordnete Menge von Namen-Wertepaaren, dargestellt mit geschweiften Klammern. Innerhalb eines Objektes gibt es mehrere Namen-Wertepaare. Ein JSON-Array, dargestellt mit eckigen Klammern, beinhaltet dagegen indizierte Listen von Elementen gleichen oder verschiedenen Typs.
JSON Web Token	Ein auf JSON basierender Access-Token. Authentifizierungsrelevante Informationen sind im Token enthalten, wodurch Sessions nicht auf einem Server gespeichert werden müssen. Ein signiertes JWT besteht aus drei Teilen, die jeweils mit Base64 kodiert werden und durch einen Punkt getrennt sind: Header, Payload und Signature.
JWT	Siehe: JSON Web Token
Mandantschaft (Mandant)	Als Mandantschaft im datenschutzrechtlichen Sinne (z. B. §4 DSGVO) wird die für bestimmte Daten verantwortlichen Stelle bezeichnet. Die Mandantschaft stellt einen abgeschlossenen und gegenüber anderen Mandantschaften klar abgegrenzten Datenhaltungs- und Verarbeitungskontext im SchulConneX-Server dar.
Natürliche Person	Siehe: Person



Begriff	Erklärung
OAuth	Bei OAuth (Open Authorization) handelt es sich um ein offenes Sicherheitsprotokoll für die Token-basierte Autorisierung und Authentifizierung im Internet. Mit Hilfe von Open Authorization kann eine nutzende Person einer Drittanwendung erlauben, auf Daten zuzugreifen, die bei einem anderen Dienst gespeichert sind, ohne dieser Passwörter offenlegen zu müssen.
Person	Im Sinne der Schnittstelle ist eine Person ein Mensch, welcher eine Beziehung zu einer oder mehreren Schulen oder auch außerschulischen Organisationen hat und dort Träger von Rechten und Pflichten ist. Personen nehmen an einer Schule in der Regel eine Primärrolle ein (z. B. Lernende/r, Lehrkraft, gesetzliche Vertretung, Ansprechperson), können aber auch weitere Schulrollen (z. B. Schulleitung) sowie Rollen in Kursen (z. B. Hauptlehrer/in, Nebenlehrer/in) und Klassen (z. B. Klassenlehrer/in, Klassensprecher/in) einnehmen.
Personenkontext	Kontext der Person, in dem ein Dienst genutzt wird. Ein Personenkontext umfasst eine Organisation und eine Rolle der nutzenden Person in Bezug auf diese Organisation, also beispielsweise Schulleitung des Heinrich-Heine-Gymnasiums.
REST	REST steht für Representational State Transfer und beschreibt das zustandslose Übertragen von Daten zwischen Client und Server. REST bedient sich folgender Prinzipien: Adressierbarkeit, Zustandslosigkeit, Unterstützung mehrerer Datenformate und standardisierte Schnittstelle.
Rolle	Eine Rolle ist eine temporäre Funktion, die eine Person oder Organisation innerhalb eines fachlichen Kontexts ausübt. Verbunden mit einer Rolle sind in der Regel Aufgaben, Befugnisse und Verantwortungen, die sich auf den Rollenträger übertragen.
SchulConneX-Service	Ein Service, der den Vorgaben der SchulConneX-Spezifikation folgt.
Service Provider	Siehe: Dienst
Sichtbar	Ein Datensatz, der von einer Organisation freigegeben wurde (siehe: Sichtfreigabe) ist für die andere Organisation sichtbar. Demensprechend listet der Endpunkt /freigaben die Freigaben aus Sicht der freigebenden Organisation und der Endpunkt /sichtbar die Freigaben aus der Sicht der Organisation, für welche die Freigabe erteilt wurde, auf.
Sichtfreigabe	Organisationen können individuelle Datensätze für andere Organisationen freigeben. Bei einer Sichtfreigabe ist für die andere Organisation nur der lesende Zugriff auf die Daten möglich. Der SchulConneX-Service unterstützt derzeit als Freigaben nur Sichtfreigaben.
XBildung (Datenmodell)	XBildung ist ein vom Bundesministerium für Bildung und Forschung und dem Finanzministerium Sachsen-Anhalt veröffentlichtes Spezifikationsdokument zur Schaffung und Wahrung von Interoperabilität für das Bildungswesen in Deutschland. Die XBildung-Spezifikation enthält einen Basissatz semantischer und technischer Spezifikationen aus dem Bildungsbereich. Fachspezifische Zusatzdokumente wie XHochschule, XSchule oder XBerufsausbildung ergänzen die Spezifikation für die jeweiligen Bereiche.

Begriff	Erklärung
XSchule	XSchule stellt ein schulbezogenes Fachmodul der XBildung Spezifikation dar und ergänzt diese Spezifikation um schulspezifische Elemente.

## 14 English API Notes (eng.)

While the overall API description is written in German, this section describes the main data models and their attributes in English. The intention of the section is to allow English speaking implementers to understand the basic ideas behind most of the API elements and attributes. While it is, presumably, not as useful as a full translation of the API document, it provides, at least, a starting point for implementation work.

Most of the descriptive texts are probably readable and understandable when run through an automatic online translation service, but the specific meaning of the attributes, especially if the attribute name is a shortening of a word, requires a more precise translation in order to be usable for implementation. Providing a short description of these terms in the context of the API is the primary purpose of this section.

### 14.1 Data Models (eng.)

#### 14.1.1 Format of dates (eng.)

Dates are always formatted as YYYY-MM-DD.

#### 14.1.2 Organisation (eng.)

An organisation is in most cases a school or a similar relevant organisation in an educational context.

Attribute	Description
<b>id</b>	ID of the organisation Provided by the server and unique
<b>kennung</b>	Optional alternative ID of an organisation (in most cases, this will be a national or regional school ID) It needs to be unique within the organisation type
<b>name</b>	Official name of the organisation

Attribute	Description
<b>namensergaenzung</b>	Addition to the organisation name (such as a previous name or a suffix that isn't part of the name)
<b>kuerzel</b>	Short name of the organisation
<b>anschrift</b>	Address of an organisation
<b>typ</b>	Organisation type. See code list Organisationstyp
<b>traegerschaft</b>	Owner type of the organisation. See code list 12.8 Trägerschaft

#### 14.1.3 Person (eng.)

Attribute	Description
<b>id</b>	ID of the person Provided by the server and unique
<b>referrer</b>	"referrer" is the ID that the local client system uses. It is used to map the ID provided by the registry server to the data stored in the local school management software.
<b>mandant</b>	ID of the organisation the person is attached to. This is mainly a legal relationship, in the sense that this is the controller responsible for maintaining the data for that person as described in DSGVO Art. 4 Nr. 7. Persons can be attached contextually to other organisations with the 'Personenkontext'.
<b>stammorganisation</b>	A person can belong administratively to one organisation, but might be delegated to another organisation. The 'stammorganisation' (home organisation) is the ID of the organisation the person primarily belongs to.
<b>name</b>	Name entity
<b>name.familienname</b>	Surname(s) of the person

Attribute	Description
<b>name.vorname</b>	Given name(s) of the person
<b>name.initialenfamiliename</b>	Initial or initials of the surname
<b>name.initialenvorname</b>	Initial or initials of the given names
<b>name.rufname</b>	Name the person is commonly called by.
<b>name.titel</b>	Title of the person
<b>name.anrede</b>	List of forms of address (e. g. Mr.)
<b>name.namenssuffix</b>	List of name suffixes
<b>name.sortierindex</b>	Sort index, denoting the index of the letter of the last name by which a name is to be sorted. The index is zero-based, so a "0" denotes sorting by the first letter of the last name.
<b>kommunikation</b>	Array of the data type 5.10 Kommunikation describing how the person can be contacted
<b>geburt</b>	Birth information entity
<b>geburt.datum</b>	Date of birth
<b>geburt.geburtsort</b>	Place of birth, usually town and country If no country is provided, Germany is assumed.
<b>geschlecht</b>	Sex of person Refer to code list 12.2 Geschlecht
<b>lokalisierung</b>	Localisation according to RFC5646 If none is provided „de“ is assumed. Refer to code list 12.9 Lokalisierung.
<b>vertrauensstufe</b>	Describes the level of trust in the data provided Refer to code list 12.4 Vertrauensstufe.

Attribute	Description
<b>auskunftssperre</b>	<p>Blocked information</p> <p>Is set to "JA" (yes) if access to information of this person is blocked</p> <p>Refer to code list 12.7</p> <p>Boolean</p>
<b>revision</b>	<p>Revision number of the information</p> <p>This is mainly used to check on updates and deletes of data sets, whether there have been any changes to the data since it was retrieved by the client system.</p>

#### 14.1.4 Personenkontext (eng.)

The person context describes the role of a person within an organisation.

A person can have multiple person contexts. These can be in the same organisation the person is attached to, but also in different organisations. However, it is not permitted to have the same person context (same role in the same organisation) twice.

Attribute	Description
<b>id</b>	ID of the person context Provided by the server and unique
<b>referrer</b>	"referrer" is the ID that the local client system uses. It is used to map the ID provided by the registry server to the data stored in the local school management software.
<b>mandant</b>	ID of the organisation the person context is attached to
<b>organisation</b>	Organisation the person has a role in See 14.1.2 Organisation
<b>rolle</b>	Role the person has in the organisation Refer to code list 12.3 Rolle
<b>personenstatus</b>	Status the person has in relation to a role See code list 12.1 Personenstatus
<b>jahrgangsstufe</b>	School grade or form a person is currently in or associated with Refer to code list 12.6 Jahrgangsstufe
<b>sichtfreigabe</b>	Boolean value that denotes whether the data is only visible due to another organisation providing read access
<b>loeschung.zeitpunkt</b>	Date and time when the person context is to be deleted
<b>revision</b>	Revision number of the information

	This is mainly used to check on updates and deletes of data sets, whether there has been any changes to the data since it was retrieved by the client system.
--	---

#### 14.1.5 Personendatensatz (eng.)

A "Personendatensatz" is essentially a wrapper object containing data about a person and the associated person contexts of that person.

Attribute	Description
<b>person</b>	Person. (See 5.3 Personen)
<b>personenkontexte</b>	Array of person contexts (see 5.4 Personenkontext)

#### 14.1.6 Gruppe (eng.)

A group (Gruppe) allows it to handle a number of persons as a group. In most cases this is a school class or similar organisational structure.

Attribute	Description
<b>id</b>	ID of the group Provided by the server and unique
<b>mandant</b>	ID of the organisation that owns the group dataset
<b>orgid</b>	ID of the organisation that the group is attached to Often the same as "mandant", but not always so
<b>referrer</b>	Referrer is the ID that the local client system uses. It is used to map the ID provided by the registry server to the data stored in the local school management software.
<b>bezeichnung</b>	Describes the group

<b>thema</b>	Keywords describing the theme of the group, short text, ideally less than 35 characters
<b>beschreibung</b>	Additional optional description of the purpose of the group
<b>typ</b>	Type of the group. Refer to code list 12.13 Gruppentyp
<b>Bereich</b>	Additional group qualifier (mandatory, optional, basic, advanced...) Refer to code list 12.10 Gruppenbereich
<b>Optionen</b>	Additional group options Usually referring to language used (bilingual, native language) Refer to code list 12.12 Gruppenoption
<b>differenzierung</b>	Specific differentiation in some educational branches Refers to code list 12.11 Gruppendifferenzierung
<b>bildungsziele</b>	Educational branch. Refers to code list 12.18 Bildungsziel
<b>jahrgangsstufen</b>	Form or course level Refers to code list 12.6 Jahrgangsstufe
<b>faecher</b>	Array to list the topics covered in the group
<b>faecher.code</b>	Topic covered Refers to code list 12.17 Fächerkanon
<b>referenzgruppen</b>	Array of reference groups for this group  Allows the association of members in one or more other groups to this group  The aim is to simplify the handling of courses and school classes by allowing, for example, to attach a whole class to a course, without the need to assign every pupil individually.
<b>referenzgruppen.grupid</b>	ID of the reference group



<b>referenzgruppe.rollen</b>	<p>Roles in the reference group ("Gruppenrollen") to be selected for the group</p> <p>If no roles are specified, all persons are selected. (The main purpose for this filter is that a class might contain a teacher and pupils, but only the pupils from this reference group participate in another group.)</p>
<b>laufzeit</b>	<p>Defines the start and end date of a group</p> <p>Both dates can be either defined by directly specifying a date or, alternatively, by associating a teaching period (like a year or a semester) with the group</p>
<b>laufzeit.von</b>	Start date of the group as a date
<b>laufzeit.vonlernperiode</b>	<p>Start date of the group as a reference to a teaching period</p> <p>Refer to code list 12.15 Lernperiode</p>
<b>laufzeit.bis</b>	End date of the group as a date
<b>laufzeit.bislernperiode</b>	<p>End date of the group as a reference to a teaching period</p> <p>Refer to code list 12.15 Lernperiode</p>
<b>sichtfreigabe</b>	Boolean value that denotes whether the data is only visible due to another organisation providing read access
<b>revision</b>	<p>Revision number of the information</p> <p>This is mainly used to check on updates and deletes of data sets, whether there have been any changes to the data since it was retrieved by the client system.</p>

#### 14.1.7 Gruppenzugehörigkeit (eng.)

The group membership connects a person context to a group. In addition to the role a person has in an organisation, an additional role in the group needs to be provided.

Attribute	Description
<b>id</b>	ID of the group membership Provided by the server and unique
<b>mandant</b>	ID of the organisation that owns the group dataset
<b>referrer</b>	Referrer is the ID that the local client system uses It is used to map the ID provided by the registry server to the data stored in the local school management software.
<b>ktid</b>	ID of the person context associated with the group
<b>rollen</b>	Role of a person within the group Refer to code list 12.14 Gruppenrolle
<b>von</b>	Begin of the group membership
<b>bis</b>	End of the group membership
<b>revision</b>	Revision number of the information This is mainly used to check on updates and deletes of data sets, whether there have been any changes to the data since it was retrieved by the client system.

#### 14.1.8 Gruppendatensatz (eng.)

The "group data set" is a wrapper object to link a group to a number of group memberships.

Attribute	Description
<b>gruppe</b>	Group (See 5.6 Gruppe)
<b>gruppenzugehoerigkeiten</b>	Array of group memberships (See 5.7 Gruppenzugehörigkeit)

#### 14.1.9 Lernperiode (eng.)

A teaching period ("Lernperiode") is a time period in which organisations can offer groups. In most cases these periods are school years or semesters.

Teaching periods are not modifiable via APIs, they are provided in the same manner as code lists. However, as they are not simple lists, but structured objects, their structure is described here.

Attribute	Description
<b>code</b>	Code of the teaching period
<b>bezeichnung</b>	Description of teaching period, for example „Schuljahr 2023/2024“
<b>typ</b>	Type of the teaching period. Refers to codelist 12.16 Lernperiodentyp
<b>beginn</b>	Beginning date of teaching period
<b>ende</b>	Ending date of teaching period

#### 14.1.10 Kommunikation (eng.)

This describes a communication channel that can be used to contact a person or an organisation.

Attribute	Description
<b>id</b>	ID of the communication Provided by the server and unique
<b>typ</b>	Type of communication channel, such as e-mail or mobile phone References code list 12.19 Erreichbarkeitstyp
<b>kennung</b>	Specific contact information for the channel specified above, for example an e-mail address or a phone number

#### 14.1.11 Beziehung (eng.)

This describes a relationship between two persons (like one person being the legal guardian, carer or tutor of another). The relation connects to person contexts.

Relationships are of the "has a" kind. If a person context of person A has a relationship of "SchB" with a person context (ktid) of person B, then A has B as a "Schulbegleiter".

Attribute	Bemerkung
id	ID of the relation
ktid	ID of the person context to the person who has the relationship  For example, if the relation is 'legal guardian', then the person with the current person context is the person with the legal guardian, while the person with the person context with the ktid is the legal guardian.
beziehung	Relations. Refers to code list 12.20 Beziehungen
revision	Revision number of the information  This is mainly used to check on updates and deletes of data sets, whether there have been any changes to the data since it was retrieved by the client system.

#### 14.1.12 Anschrift (eng.)

This is the address (physical location) of, for example, an organisation.

Attribut	Bemerkung
postleitzahl	ZIP Code
ort	Town name
ortsteil	Urban district, if a more specific location than town name is required

#### 14.1.13 Organisationsbeziehung (eng.)

Describes the relationship of two organisations.

Attribut	Bemerkung
<b>orgid</b>	ID of the organisation the current organisation has a „has as“ relation.
<b>organisationsbeziehung</b>	Type of relation between the organisations

## 14.2 Code lists (eng.)

The API uses a number of code lists for many of the attributes, to ensure that attribute values are consistent and that implementations do not use different representations for the same data.

All codes in code lists are considered case-insensitive, so SCHULE", "schule", "Schule" and "ScHuLe" are considered equivalent and valid.

As most of the codes map to meanings that are reflected in the GUI of software used in schools, translating the meaning of the individual codes will likely not be of much use. For providing the GUI of a tool used in schools, it will be necessary to work with someone who has a good understanding of the German language.

However, it is not necessary to understand the meaning of the individual codes to be able to use the API itself. To provide some basic idea of the purpose of the individual code lists, a short description of all lists is provided, though a translation of the individual codes is not given.

Code list name	Description
<b>Personenstatus</b>	This is used to describe whether the dataset for a person is currently active
<b>Geschlecht</b>	Options for the sex of a person (male/female/divers/not-given)
<b>Rolle</b>	The role within an organisation, such as student, teacher or administrator
<b>Vertrauensstufe</b>	Verification level of the owner of a (digital access) key, from untrusted or unknown to fully trusted
<b>Organisationstyp</b>	Type of an organisation Usually a school, but can be an external provider of educational content.
<b>Jahrgangsstufe</b>	Describes which form or level (currently year 1 to 10) a pupil is in and for which a course is suitable for
<b>Boolean</b>	Defines what terms are used for boolean values ("ja" for true and "nein" for false)
<b>Trägerschaft</b>	The 'ownership' of an organisation Describes whether it is owned by the state, the county, the church or privately owned
<b>Lokalisierung</b>	Provides localisation information Usually "de" to denote German language content
<b>Gruppenbereich</b>	Denotes whether participation in a group is optional ("Wahl"), mandatory ("Pflicht") or a combination of both for its members

<b>Gruppendifferenzierung</b>	An attribute to specify a specific level of a course  Conceptually similar to denoting courses in the UK education system as "A-Level" or "O-Level" courses
<b>Gruppenoption</b>	Currently there are only two options available, one being "bilingual" and the other "taught in (students's) native language"
<b>Gruppentyp</b>	Classification of a group as "class", "course" or "other"  Classes are generally mandatory, while courses are often optional
<b>Gruppenrolle</b>	Similar to "Rolle", but as this refers to roles in a group and not in an organisation, the options differ slightly  For example, groups cannot have system administrator, while organisations don't have group leaders.
<b>Lernperiode</b>	A teaching period, usually a year or a semester (trimesters are rare in Germany)  A more specific description of the structure of codes from this list is given in 14.1.9 Lernperiode.
<b>Lernperiodentyp</b>	Type (essentially the length) of a teaching period  Year or half-year
<b>Fächerkanon</b>	Subjects that can be taught in a class or course  Such as biology, French or natural sciences
<b>Bildungsziel</b>	Type of school, based on likely qualification level upon leaving
<b>Erreichbarkeitstyp</b>	Type of a channel that can be used to contact a person or organisation
<b>Beziehungen</b>	Relations that can exist between two people, such as legal guardian, tutor or carer
<b>Organisationsbeziehungen</b>	Relations between organisations, often between a school and its supervising authority

## 15 Praxisleitfaden für die Implementierung

Quellsysteme haben die Möglichkeit, das Quellsysteme-API des SchulConneX-Servers zu nutzen und darüber Daten mit einem SchulConneX-Service zu synchronisieren.

Bei denen in diesem Kapitel gegebenen Ablaufbeschreibungen handelt es sich um Empfehlungen. Das Kapitel ist nicht Teil der verbindlichen Spezifikation.

### 15.1 Ablaufbeschreibung: Synchronisation mit dem primären Quellsystem

Diese Sektion beschreibt den vorgeschlagenen grundsätzlichen Ablauf der Synchronisation zwischen einem primären Quellsystem (Stammdatenverwaltung des Schulverwaltungssystems) und des SchulConneX-Services. Aus dem Vorschlag lassen sich die notwendigen Teile der Schnittstellenimplementierung auf Seiten des Quellsystems ableiten.

#### 15.1.1 Konfiguration

Das Quellsystem muss sich für die Nutzung der REST-API des SchulConneX-Services authentifizieren. Bei Verwendung des OAuth2-Flows werden dafür der Password Credentials Grant verwendet. Das Quellsystem erhält durch die Authentifizierung einen Access Token, mit dem die Endpunkte der REST-API aufgerufen werden können.

Das Quellsystem muss die Zugangsdaten (Kennung, Passwort, ggf. noch URLs) sicher speichern können. Eine lokale Speicherung oder Zwischenspeicherung (Datei) ist nicht zulässig. Es muss sichergestellt sein, dass nur berechtigte Nutzende Zugriff auf das Client Secret haben bzw. Funktionen aufrufen können, die das Client Secret verwenden. Falls die Funktion von einem lokalen Rechner ausgelöst wird, ist sicherzustellen, dass das Client Secret nicht von Unberechtigten ausgelesen werden kann.

Meistens haben Schulverwaltungssysteme ein Rechte- und Rollensystem zur Steuerung des Zugriffs auf einzelne Funktionalitäten. Auch das Recht, den Synchronisationsvorgang auszulösen, sollte an eine Rolle oder Benutzer/innen gebunden sein. Diese Entscheidung liegt jedoch beim Betreiber oder Hersteller des Schulverwaltungssystems.

#### *Anforderungen an das Quellsystem*

- Implementierung eines Unterpunkts im Konfigurationsmenü zur Eingabe und zum Abspeichern der Zugangsdaten
- Implementierung eines Rechts "Synchronisation mit SchulConneX-Service" und der Möglichkeit, dieses Recht an Rollen oder Benutzer oder Benutzerinnen zu vergeben.

#### 15.1.2 Synchronisation

Der Synchronisationsvorgang besteht aus mehreren Schritten.

1. Auslösen des Synchronisationsvorgangs
2. Auswahl der zu synchronisierenden Gruppen
3. Abruf der Personen des Mandanten des Quellsystems aus dem SchulConneX-Service
4. Abgleich der Datensätze
5. Aktualisieren oder Anlegen der jeweiligen Datensätze



6. Importieren bestätigter Datensätze in das Quellsystem
7. Abruf der Gruppen des Mandanten des Quellsystems aus dem SchulConneX-Service
8. Abgleich der Datensätze (Gruppen und Gruppenzugehörigkeiten)
9. Aktualisieren und Anlegen der Datensätze (Gruppen und Gruppenzugehörigkeiten)
10. Importieren bestätigter Datensätze (Gruppen und Gruppenzugehörigkeiten) in das Quellsystem
11. Rückmeldung an den Benutzer oder die Benutzerin

Diese Schritte, sowie teilweise die daraus resultierenden Anforderungen an ein Quellsystem, werden in den folgenden Abschnitten genauer beschrieben.

### *1. Auslösen des Synchronisationsvorgangs*

Benutzer oder Benutzerinnen mit entsprechenden Rechten können diese Funktion, beispielsweise über eine Schaltfläche oder einen Menüeintrag, auslösen.

#### *Anforderungen and das Quellsystem*

Implementierung eines UI-Elements zum Auslösen des Synchronisationsvorgangs, wie einen Button oder eine Menü-Funktion.

### *2. Auswahl der zu synchronisierenden Gruppen*

Meistens ist es nicht sinnvoll, sämtliche Personeneinträge der Datenbank des Quellsystems ohne Vorauswahl zu synchronisieren. Daher sollte im 2. Schritt die Möglichkeit bestehen, die zu synchronisierenden Gruppen auszuwählen. Es bietet sich ein modaler Dialog an, mit der Möglichkeit Schulklassen oder andere einschlägige Personengruppen wie Lehrkräfte auszuwählen.

#### *Anforderungen and das Quellsystem*

- Implementierung eines Dialogs zur Auswahl von Klassen sowie der Gruppe der Lehrkräfte

### *3. Abruf der Personen des Mandanten des Quellsystems aus dem SchulConneX-Service*

Das Quellsystem sollte vor dem Erstellen des Synchronisationsdatensatzes über den Endpunkt `/personen` (oder alternativ über `/personenkontexte`) der REST-API zunächst den vollständigen Datensatz an Personen und Personenkontexten des Mandanten abrufen. Die Definition der Schnittstellen und ihre Rückgabewerte mit Beispielen sind aus der Schnittstellenspezifikation für Quellsysteme ersichtlich.

#### *Hinweise:*

Eine Person kann in mehreren Rollen im landesweiten Schulsystem vertreten sein, beispielsweise gleichzeitig als Lehrkraft und Erziehungsberechtigte oder als Schulleitung und

Lehrkraft. Daher können zu einer Person mehrere Personenkontexte gehören, wobei der Regelfall jedoch ein Personenkontext pro Person ist.

#### Anforderungen and das Quellsystem

- Implementierung der Abruffunktion des REST-Endpunkts `/personen`

#### 4. Abgleich der Datensätze

Der Abgleich der Datensätze erfolgt anhand der Schlüssel. Hier kann die UUID des SchulConneX-Services verwendet werden. In diesem Fall muss das Quellsystem die UUID aus dem SchulConneX-Service als Fremdschlüssel selbst speichern. Alternativ kann das Quellsystem seinen eigenen Schlüssel als Fremdschlüssel ("referrer") im SchulConneX-Service ablegen und den Abgleich darüber durchführen.

Falls beide Möglichkeiten genutzt werden, sollte das Quellsystem eine Strategie festlegen, wie bei Schlüsselkonflikten verfahren wird. Der SchulConneX-Service verwendet immer den eigenen Schlüssel. Das Quellsystem muss eine Strategie festlegen, wie mit den Schlüsseln umgegangen wird, weil das Quellsystem seinen eigenen Schlüssel hat und die UUID des SchulConneX-Services.

Fall	Beschreibung	Aktion
1	Der Datensatz im SchulConneX-Service kann einem Datensatz im Quellsystem über einen der beiden Schlüssel eindeutig zugeordnet werden.	Vormerken für automatische Aktualisierung
2	Sowohl im Datensatz im SchulConneX-Service als auch im Datensatz des Quellsystems sind sämtliche Personenattribute (Familiename, Vorname, Geburt: Datum, Ort) vorhanden. Die Attributwerte stimmen sämtlich überein und eine Zuordnung ist daher auch ohne Schlüssel eindeutig möglich.	Vormerken für automatische Aktualisierung
3	Im Datensatz SchulConneX-Service oder im Datensatz Quellsystem sind nur ein Teil der Personenattribute (Familiename, Vorname, Geburt: Datum, Ort) vorhanden. Die Attributwerte stimmen überein, aber die Zuordnung ist möglicherweise nicht eindeutig.	Vormerken für Aktualisierung nach Zuordnungs-Bestätigung
4	Der Datensatz im SchulConneX-Service hat einen Fremdschlüssel (referrer) des Quellsystems, ist aber anhand des Schlüssels im Quellsystem nicht auffindbar. (Dieser Fall kann auftreten, wenn im Quellsystem ein Backup wiedereingespielt wurde,	Vormerken für Import nach Import-Bestätigung

	nachdem das Quellsystem bereits neue Personen im SchulConneX-Service hat.)	
5	Der Datensatz im SchulConneX-Service kann einem Datensatz im Quellsystem nicht zugeordnet werden.	Vormerken für Import nach Import-Bestätigung
6	Nach Abgleichen der Fälle 1, 2, 3 und 4 kann ein Datensatz im Quellsystem nicht einem Datensatz im SchulConneX-Service zugeordnet werden.	Vormerken zum automatischen Anlegen eines neuen Datensatzes im SchulConneX-Service

Tabelle 46: Ablaufbeschreibung für den Abgleich der Datensätze

#### Anforderungen an das Quellsystem

- Implementierung der Geschäftslogik zum Abgleichen der Datensätze
- Implementierung eines Dialogs zum Auswählen (Bestätigen) im Fall 3
- Implementierung eines Dialogs zum Auswählen (Bestätigen) in den Fällen 4 und 5 (Import)

*Die Möglichkeit, unbekannte Datensätze aus dem SchulConneX-Service in das Quellsystem zu importieren (Fälle 4 und 5) kann auch für eine spätere Ausbaustufe zurückgestellt werden.*

#### 5. Aktualisieren und Anlegen der Datensätze

Das Quellsystem hat nach Abgleich der Datensätze und eventuell notwendigen Bestätigungen (Fälle 3, 4 und 5) je eine Liste von Datensätzen zum Aktualisieren und zum Anlegen. Das Aktualisieren und Anlegen von Personen und Personenkontexten geschieht über getrennte Endpunkte.

Aktion	Endpunkte	Hinweis
Aktualisieren	PUT /personen/{id} PUT /personenkontexte/{id}	Die Revision muss als Attribut mitgegeben werden.
Anlegen	POST /personen/ POST /personen/{id}/personenkontexte/	

Tabelle 47: Ablaufbeschreibung für das Aktualisieren und Anlegen der Datensätze

Eine PATCH-Operation ist nicht vorgesehen. Beim Aktualisieren über PUT muss der gesamte Datensatz mitgegeben werden. Es ist darauf zu achten, dass das Quellsystem dabei nicht unbeabsichtigt Daten löscht.

#### *6. Importieren bestätigter Datensätze in das Quellsystem*

Wenn (in den Fällen 4 und 5) dem Import zugestimmt wurde und die Attribute im SchulConneX-Service zum Anlegen eines neuen Datensatzes im Quellsystem ausreichen, kann der Import durchgeführt werden.

*Die Möglichkeit, unbekannte Datensätze in das Quellsystem zu importieren (Fälle 4 und 5), kann auch für eine spätere Ausbaustufe zurückgestellt werden.*

#### *7. Abruf der Gruppen des Mandanten des Quellsystems aus dem SchulConneX - Service*

Das Quellsystem sollte vor dem Erstellen des Synchronisationsdatensatzes über den Endpunkt `/gruppen` des REST-API zunächst den vollständigen Datensatz an Gruppen und Gruppenzugehörigkeiten des Mandanten aus dem SchulConneX-Service abrufen. Die Definition der Schnittstellen und ihre Rückgabewerte mit Beispielen sind aus der Schnittstellenspezifikation für Quellsysteme ersichtlich.

#### *8. Abgleich der Datensätze (Gruppen und Gruppenzugehörigkeiten)*

Der Abgleich der Datensätze für Gruppen und Gruppenzugehörigkeiten sollte analog zum Abgleich der Datensätze für Personen und Personenkontexte in (4) erfolgen. Dieser Import ist optional.

Fall	Beschreibung	Aktion	Hinweis
1	Der Datensatz aus dem SchulConneX-Service kann einem Datensatz im Quellsystem über einen der beiden Schlüssel eindeutig zugeordnet werden.	Vormerken für automatische Aktualisierung	
2	Der Datensatz aus dem SchulConneX-	Vormerken für Import nach	Dieser Fall kann z. B. auftreten, wenn im Quellsystem ein Backup wiedereingespielt wurde, nachdem das Quellsystem bereits

	Service hat einen Fremdschlüssel (referrer) des Quellsystems, ist aber anhand des Schlüssels im Quellsystem nicht auffindbar.	Import-Bestätigung	neue Gruppen im SchulConneX-Service angelegt hat.
3	Der Datensatz aus dem SchulConneX-Service kann einem Datensatz im Quellsystem nicht zugeordnet werden.	Vormerken für Import nach Import-Bestätigung	Dieser Fall tritt auf, wenn Gruppen in der Administrationsoberfläche des SchulConneX-Services manuell oder durch Synchronisation mit einem sekundären Quellsystem angelegt wurden. Ersteres kann bei Arbeitsgruppen/Teams oder ähnlichen Fällen gegeben sein, letzteres, wenn die Unterrichte zu Klassen von einer Stundenplanverwaltung oder einer Oberstufenverwaltung angelegt wurden (sekundäre Quellsysteme). Der Rückimport dieser Gruppen (Unterrichte) in das primäre Quellsystem sollte nicht automatisch passieren, sondern nur nach Rückfrage bzw. entsprechend konfigurierbarer Parameter (z. B. nur Unterrichte und mit bekannten Referenzgruppen).
4	Nach Abgleichen der Fälle 1, 2 und 3 kann ein Datensatz im Quellsystem nicht einem Datensatz aus dem SchulConneX-Service zugeordnet werden.	Vormerken zum automatischen Anlegen eines neuen Gruppen-Datensatzes im SchulConneX-Service	

Tabelle 48: Ablaufbeschreibung für den Abgleich der Datensätze (Gruppen und Gruppenzugehörigkeiten)

#### Anforderungen an das Quellsystem

- Implementierung der Geschäftslogik zum Abgleichen der Datensätze

- Implementierung eines Dialogs zum Auswählen (Bestätigen) im Fall 2 und 3

Die Möglichkeit, unbekannte Datensätze in das Quellsystem zu importieren (Fälle 2 und 3), ist optional.

### *9. Aktualisieren und Anlegen der Datensätze (Gruppen und Gruppenzugehörigkeiten)*

Das Quellsystem hat nach Abgleich der Datensätze und ggf. notwendigen Bestätigungen (Fälle 2 und 3) je eine Liste von Datensätzen zum Aktualisieren und zum Anlegen von Gruppen. Das Aktualisieren und Anlegen von Gruppen und Gruppenzugehörigkeiten geschieht über getrennte Endpunkte.

Aktion	Endpunkte	Hinweis
<b>Aktualisieren</b>	PUT /gruppen/{uuid} PUT /gruppenzugehoerigkeiten/{uuid}	Die Revision muss als Attribut mitgegeben werden.
<b>Anlegen</b>	POST /gruppen/ POST /gruppen/{uuid}/ gruppenzugehoerigkeiten/	Jede Gruppenzugehörigkeit muss einzeln angelegt werden.
<b>Löschen</b>	DELETE /gruppen/ DELETE /gruppenzugehoerigkeiten/{uuid}	Jede Gruppenzugehörigkeit muss einzeln gelöscht werden.

*Tabelle 49: Ablaufbeschreibung für das Aktualisieren und Anlegen der Datensätze (Gruppen und Gruppenzugehörigkeiten)*

Eine PATCH-Operation ist nicht vorgesehen. Beim Aktualisieren über PUT muss der gesamte Datensatz mitgegeben werden. Es ist darauf zu achten, dass das Quellsystem dabei nicht unbeabsichtigt Daten löscht, insbesondere dürfen nicht Daten gelöscht werden, die von anderen Quellsystemen der Organisation benötigt oder verwaltet werden.

Falls die Synchronisation im Zusammenhang oder direkt nach einem Schuljahres- oder Schulhalbjahreswechsel stattfindet, müssen Anforderungen berücksichtigt werden, die in 15.2 Ablauf des Schuljahreswechsels für Quellsysteme beschrieben sind.

### *10. Importieren bestätigter Datensätze (Gruppen und Gruppenzugehörigkeiten) in das Quellsystem*

Wenn - in den Fällen 2 und 3 – der oder die Nutzende dem Import zugestimmt hat und die Attribute im SchulConneX-Service zum Anlegen eines neuen Datensatzes im Quellsystem ausreichen, kann der Import durchgeführt werden. Dieser Import ist optional.

## 11. Rückmeldung an den Benutzer oder Benutzerin

Zum Abschluss der Synchronisation sollte eine Rückmeldung über Erfolg oder Misserfolg erfolgen, beispielsweise als Dialog oder Statusmeldung.

Idealerweise sollte hier auch die Anzahl der aktualisierten und angelegten Datensätze angezeigt werden.

### 15.2 Ablauf des Schuljahreswechsels für Quellsysteme

Im Zusammenhang mit dem Schuljahreswechsel (zum Teil auch mit dem Schulhalbjahreswechsel) muss im Allgemeinen ein großer Teil der Daten im SchulConneX-Service aktualisiert werden.

Betroffen sind beispielsweise:

- abgehende Schülerinnen und Schüler
- abgehende Lehrkräfte
- neu aufgenommene Schülerinnen und Schüler
- neu aufgenommene Lehrkräfte
- abgehende und neue Schulklassen
- Versetzung der Klassen in das neue Schuljahr
- neue Unterrichte oder Kurse des Schuljahres

Der primäre Einsatzzweck des SchulConneX-Services ist die Bereitstellung und Übermittlung von Daten an angebundene Dienste. Dazu müssen nur die Daten der aktuellen Lernperiode vorgehalten werden; das Vorhalten von „historischen“ Daten aus vergangenen Lernperioden und Planungsdaten für zukünftige Lernperioden ist nicht notwendig.

Für Quellsysteme sind keine besonderen Funktionen oder API-Endpunkte für den Wechsel der Lernperioden vorgesehen. Findet eine Datensynchronisation allerdings im Zusammenhang oder direkt nach einem Wechsel der Lernperiode statt, müssen einige Besonderheiten berücksichtigt werden. Grundsätzlich sollte der Ablauf der Datensynchronisation der gleiche wie bei einer unterjährigen Synchronisation sein. So ist unter anderem auch sichergestellt, dass nachträgliche Änderungen beispielsweise am Stundenplan und bei Lerngruppen im Zusammenhang mit der Regelsynchronisation wie bei einem Lernperiodenwechsel übernommen werden.

#### 15.2.1 Besonderheiten bei der Aktualisierung der Personen und Personenkontexte

Daten zu Personen und Personenkontexte, welche weiterhin bei der Institution verbleiben, sollten nicht gelöscht werden. Sie müssen im SchulConneX-Service mit der gleichen UUID weitergeführt werden. Das Löschen eines Personenkontexts kann sonst in allen verbundenen Diensten zum Löschen des Benutzerkontos und damit zum Datenverlust führen.

Für Personen, die das Schulsystem des Landes verlassen oder die Organisation wechseln, sind die Anforderung zur Löschung von Personen und Personenkontexten sowie die zum

Schulwechsel zu beachten. Für Personen, die zum neuen Schuljahr an die aktuelle Schule wechseln, sind ebenfalls die Anforderungen zum Schulwechsel zu beachten.

Bei Lernenden (Schülerinnen und Schülern) muss meist das Feld "personenkontext.jahrgangsstufe" aktualisiert werden.

#### 15.2.2 Besonderheiten beim Aktualisieren von Gruppen

Gruppen, die über den Schuljahreswechsel hinweg fortbestehen, sollen auf keinen Fall gelöscht werden. Sie sollen im SchulConneX-Service mit der gleichen UUID weitergeführt werden. Auch das Löschen von Gruppen kann bei verbundenen Diensten zu Datenverlust oder zu Dateninkonsistenzen führen. So können in einem Lernmanagementsystem an einen mehrjährigen Kurs Unterrichtsmaterialien oder Aufgaben gebunden sein, auf die auch im Folgeschuljahr noch zurückgegriffen werden können muss.

Demzufolge muss das Quellsystem bei der Aktualisierung von Gruppen zwischen fortgeführten und beendeten Gruppen unterscheiden.

#### Fortgeführte Gruppen

Bei fortgeführten Gruppen werden die Attribute der Gruppe und auch gegebenenfalls die Gruppenzugehörigkeiten aktualisiert. Die Gruppe wird im SchulConneX-Service nicht gelöscht und nicht neu angelegt. Die bestehende UUID bleibt erhalten.

- Bei Schulklassen, Kursen und Unterrichten muss meist das Attribut "Bezeichnung" aktualisiert werden. Die Regeln der Aktualisierung hängen von der Datenhaltung des Quellsystems und dem Benennungsschema der Schule ab.
- Bei Schulklassen, Kursen und Unterrichten muss meist das Attribut "Jahrgangsstufen" aktualisiert werden.
- Die Attributstruktur "Laufzeit" muss überprüft und eventuell angepasst werden.

#### Beispiele:

Gruppentyp	Bezeichnung im abgeschlossenen Schuljahr	Aktualisierte Bezeichnung
Schulklasse	"Klasse 7a"	"Klasse 8a"
Schulklasse	"Qualifikationsphase 1"	"Qualifikationsphase 2"
Kurs / Unterricht	"LK Mathematik Q1"	"LK Mathematik Q1"
Kurs / Unterricht	"LK Mathematik Abitur 2024"	nicht aktualisiert
Kurs / Unterricht	"Deutsch 7a"	"Deutsch 8a"
Kurs / Unterricht	"Spanisch 9 I"	"Spanisch 10 I"
Kurs / Unterricht	"AG Basketball"	nicht aktualisiert



Sonstige Gruppe	"Kollegium"	nicht aktualisiert
Sonstige Gruppe	"Arbeitsgruppe Schulentwicklung"	nicht aktualisiert

Tabelle 50: Besonderheiten beim Aktualisieren fortgeführter Gruppen

### Beendete Gruppen

Für Gruppen, die am Ende einer Lernperiode nicht fortgeführt werden, gibt es zwei Möglichkeiten

1. sofortiges Löschen im SchulConneX-Service
2. Aktualisierung der Attributstruktur "Laufzeit" und Löschung nach einer bestimmten Zeit (beispielsweise 30 Tage). Dies hat den Vorteil, dass die verbundenen Dienste über die Beendigung der Gruppe noch informiert werden. Lehrkräfte können dann noch für den vorgegebenen Zeitraum auf Ergebnisse des vergangenen Schuljahres zurückgreifen.

Es gibt keinen klassischen Schuljahreswechsel im SchulConneX-Service, bei dem ein neues „leeres“ Schuljahr angelegt wird. Alle bestehenden Gruppen müssen explizit aktualisiert oder gelöscht werden. Es gibt keinen automatischen Löschvorgang beim Schuljahreswechsel.

### Beispiele:

Gruppentyp	Bezeichnung im abgeschlossenen Schuljahr	Erläuterung
Schulklasse	"Qualifikationsphase 2"	Klasse ist abgegangen
Kurs / Unterricht	"LK Mathematik Q2"	Kurs ist beendet
Kurs / Unterricht	"Deutsch 8a"	Bei Wechsel der Lehrkraft wird "Deutsch 9a" mit neuer UUID neu angelegt.
Kurs / Unterricht	"Physik 7a"	kein Physikunterricht im 8. Jahrgang
Kurs / Unterricht	"AG Fußball"	wenn die Schule AGs grundsätzlich nicht als "fortgeführt" ansieht

Tabelle 51: Besonderheiten beim Aktualisieren beendeter Gruppen

Die Beispiele zeigen, dass es nicht in allen Fällen verbindliche Angaben darüber geben kann, wann und ob eine Gruppe als fortgeführt oder beendet gilt. So kann "Deutsch 7a" als fortgeführt in "Deutsch 8a" angesehen werden. In diesem Fall bleiben die UUID und damit

alle in verbundenen Diensten vorhandenen Daten dieser Gruppe erhalten. Im zweiten Beispiel würde "Deutsch 8a" gelöscht (oder es würde zunächst das Laufzeitende auf das Ende des abgeschlossenen Schuljahres gesetzt werden). Danach würde eine neue Gruppe "Deutsch 9a" angelegt werden. Dies kann erwünscht sein, wenn alle Daten der Gruppe im verbundenen Lernmanagementsystem (Dateien, Aufgaben, Abgaben) gelöscht werden sollen, beispielsweise beim Wechsel der Lehrkraft.

#### **Orientierung:**

- Klassen sollen fortgeführt werden.
- Oberstufenkurse sollen fortgeführt werden.
- Kurse und Unterrichte sollen fortgeführt werden, wenn
  - sich die Gruppenzusammensetzung nicht ändert,
  - das Fach im Folgeschuljahr unterrichtet wird
  - und sich die Lehrkraft nicht ändert.

Das Quellsystem sollte der Schule per allgemeiner Konfiguration oder bei jedem Aufruf der Synchronisationsfunktion Optionen zur Fortführung geben, beispielsweise in der Form:

- ☐ Klassen fortführen
  - ☐ Kurse und Unterrichte (Oberstufe) fortführen
    - ☐ immer
    - ☐ bei gleicher Lehrkraft
  - ☐ Kurse und Unterrichte (SEK I) fortführen
    - ☐ immer
    - ☐ bei gleicher Lehrkraft
- ☐ beendete Gruppen erst nach 30 Tagen löschen

Ein Quellsystem sollte beim Synchronisationsvorgang zwischen den Schritten „Abgleich der Gruppen-Datensätze mit dem SchulConneX-Service" und „Aktualisieren, Löschen oder Neuanlegen der Gruppen-Datensätze" immer einen Dialog zeigen, in dem die zu synchronisierenden Gruppen aufgelistet werden und die geplante noch einmal bestätigt werden muss oder manuell einzeln geändert werden kann. Auch die oben erwähnte Löschfrist sollte einzeln änderbar sein.

### 15.3 Änderung der Rolle in einer Organisation

Technisch ist es derzeit möglich, die Rolle innerhalb eines Personenkontextes über die API Schnittstelle `/personenkontexte/{id}` direkt zu ändern.

In Implementierungen ist eine solche direkte Änderung zu vermeiden.

Der empfohlene Ablauf ist wie folgt:

1. Erstellen eines neuen Personenkontextes mit der neuen Rolle
2. Überprüfen der erfolgreichen Erstellung des neuen Personenkontextes
3. Löschen des alten Personenkontextes

Grund für dieses Vorgehen ist das Vermeiden potentieller Fehler.

- Es darf für eine Person keine zwei Personenkontexte mit identischen Rollen in einer Organisation geben. Bei direkter Änderung könnte es geschehen, dass ein Konflikt mit einem bereits bestehenden Kontext entsteht. Dieser Fehler kann durch schrittweise Änderung (erst erstellen, dann löschen) klarer erkannt und einfacher behandelt werden.
- Durch die Erstellung eines neuen Personenkontextes vor dem Löschen des alten Personenkontextes wird vermieden, dass es zu Rechte-Problemen durch die Änderungsreihenfolge kommt. Insbesondere wird vermieden, dass bei einer Änderung von der Administratorrolle in eine Rolle mit weniger Berechtigungen die Löschung der Administratorrolle erfolgt, aber in der Folge das Recht zur Erstellung der anderen Rolle nicht mehr besteht.
- Durch die explizite Löschung des ursprünglichen Personenkontextes und Neuerstellung eines anderen Personenkontextes, statt einer direkten Änderung, wird sichergestellt, dass die Mechanismen zur Benachrichtigung anderer Dienste (wie das Löschedatum oder das Erstellungsdatum der Personenkontext-ID) funktionieren.
- Es ist möglich, dass eine spätere Version dieser Spezifikation die direkte Änderung des Personenkontextes ausschließen wird, so dass die hier beschriebene Vorgehensweise die Zukunftssicherheit der Anwendung erhöht.

## 16 Anlage: Dokumenthistorie

Datum	Autor*in	Organisation	Version	Status	Änderungen
22.03.2022	Christopher Krauss	Fraunhofer FOKUS	0.000.001	Interner Arbeitsentwurf	Vorlage erstellt
22.03.2022 – 06.05.2022	Truong-Sinh An, Christoph Müller, Christian Fuhrhop, Görkem Güclü, André Paul, Christopher Krauss	Fraunhofer FOKUS	0.000.002 - 0.000.004	Interner Arbeitsentwurf	Ausarbeitung einer minimalen Spezifikation für Service Provider
06.05.2022	Christopher Krauss	Fraunhofer FOKUS	0.001.000	Kommentierbarer Arbeitsentwurf	Bereitstellung des Entwurfs zur Kommentierung durch PO, QS und Dev-Team
06.05.2022 – 13.05.2022	Boris Heithecker und Harald Dahlen	n-21, topdev	0.001.001	Kommentierbarer Arbeitsentwurf	Kommentierung/ Qualitätssicherung
13.05.2022 – 20.05.2022	Christoph Müller, Christian Fuhrhop, Görkem Güclü, André Paul, Christopher Krauss	Fraunhofer FOKUS	0.001.002	Kommentierbarer Arbeitsentwurf	Einarbeiten und Auflösen der Kommentare. Treffen finaler Festlegungen.
20.05.2022	Christopher Krauss	Fraunhofer FOKUS	0.001.003	Veröffentlichungskandidat	Bereitstellung des Veröffentlichungskandidaten der Spezifikation für Service Provider
04.07.2022 – 08.07.2022	Christopher Krauss	Fraunhofer FOKUS	0.001.004	Veröffentlichungskandidat	Einarbeiten von Feedback erster Dienste und Korrekturen, Gendern
14.07.2022 – 15.07.2022	Boris Heithecker, Harald Dahlen, Christopher Krauss	n-21, topdev, FOKUS	1.000.000	Veröffentlichte Spezifikation	Finale Qualitätskontrolle und Finalisierung der Spezifikation in Version 1.000.000
22.07.2022 – 25.07.2022	Christian Fuhrhop	Fraunhofer FOKUS	1.010.000	Veröffentlichte Spezifikation	Erste (Wieder-)Zusammenführung der Teilspecs und der Code-Tabellen zu einem gemeinsamen Dokument
02.08.2022	Christian Fuhrhop	Fraunhofer FOKUS	1.010.010	Veröffentlichte Spezifikation	Einarbeitung offener Kommentare

08.08.2022- 10.08.2022	Christian Fuhrhop	Fraunhofer FOKUS	1.011.100	Kommentierbarer Arbeitsentwurf	Ergänzung mit Gruppen und gruppenspezifischen APIs
15.08.2022- 16.08.2022	Christian Fuhrhop	Fraunhofer FOKUS	1.012.100	Interner Arbeitsentwurf	Ergänzung mit Gruppenzugehörigkeit und Gruppenzugehörigkeit-APIs
17.08.2022- 18.08.2022	Christian Fuhrhop	Fraunhofer FOKUS	1.013.100	Interner Arbeitsentwurf	Entfernung Ergänzung mit Lernperioden und Lernperioden-APIs
19.08.2022	Christian Fuhrhop	Fraunhofer FOKUS	1.014.100	Interner Arbeitsentwurf	Entfernung der Gruppenzugehörigkeit-APIs bei Personen, Erweiterung der Gruppenzugehörigkeit um Referrer, Definition Datumsformat
22.08.2022	Christian Fuhrhop	Fraunhofer FOKUS	1.015.100	Interner Arbeitsentwurf	Update Inhalt API Response für Gruppen
24.08.2022 – 25.08.2022	Christian Fuhrhop	Fraunhofer FOKUS	1.016.100	Interner Arbeitsentwurf	Umschreiben diverser Gruppen und Gruppenzugehörigkeits- Attribute und Strukturen
26.08.2022	Christian Fuhrhop	Fraunhofer FOKUS	1.017.100	Interner Arbeitsentwurf	Anpassung "Option" auf "Optionen", Einfügung aktueller Fächerkatalog
31.08.2022	Christian Fuhrhop	Fraunhofer FOKUS	1.018.100	Interner Arbeitsentwurf	QS-Gruppen und Gesamtdokument
08.09.2022	Christian Fuhrhop	Fraunhofer FOKUS	1.019.100	Interner Arbeitsentwurf	Update Codelisten, Anpassung Initialen, Codeliste Gruppenrollen
08.09.2022	Christian Fuhrhop	Fraunhofer FOKUS	1.020.100	Interner Arbeitsentwurf	Update Gruppenzugehörigkeiten APIs um Gruppeninformationen (Gruppendatensatz)
12.09.2022	Christian Fuhrhop	Fraunhofer FOKUS	1.021.100	Interner Arbeitsentwurf	API /gruppenzugehörigkeiten. Anpassung Gruppenrollen, Hinweis auf Niedersachsen- Spezifität einiger Codelisten, Hinweis zu Vorhandensein von Attributen bei Freigabe oder fehlender Freigabe, Anpassung von Umfang Gruppendatensatz für die APIs, Hinzufügen von Initialen (Vorname/Nachname) zum Quellsystem-Personen- Datenmodell.

15.09.2022	Christian Fuhrhop	Fraunhofer FOKUS	1.022.100	Interner Arbeitsentwurf	Filterfunktionen für Gruppen und Gruppenzugehörigkeiten APIs
15.09.2022	Christopher Krauß	Fraunhofer FOKUS	1.023.100	Kommentierbarer Arbeitsentwurf	Formatierung und Freigabe als kommentierbarer Arbeitsentwurf.
23.09.2022	Christian Fuhrhop	Fraunhofer FOKUS	1.024.100	Kommentierbarer Arbeitsentwurf	Angepasste Organisation für Dienstanbieter, Einarbeitung Kommentare, Anpassung von Referenzen auf Word-Cross-Referenzen
28.09.2022	Christian Fuhrhop	Fraunhofer FOKUS	1.025.100	Kommentierbarer Arbeitsentwurf	JSON Validate für alle JSON-Beispiele, Abgleich Beispielcodes mit Codelisten, Korrektur Nummerierung von Captions
30.09.2022	Christian Fuhrhop	Fraunhofer FOKUS	1.026.000	Kommentierbarer Arbeitsentwurf	Attribut orgid zu Gruppen-Attributen hinzugefügt
07.10.2022	Christian Fuhrhop, Peggy Krüger	Fraunhofer FOKUS	1.027.000	Kommentierbarer Arbeitsentwurf	Abarbeitung Kommentare nach Kommentierungsphase, Gruppenattribute für Dienstanbieter, Optionale Attribute und Freigabe
07.10.2022	Christian Fuhrhop	Fraunhofer FOKUS	1.028.000	Kommentierbarer Arbeitsentwurf	Abschluss Überarbeitung und Korrekturen
10.10.2022	Christopher Krauß	Fraunhofer FOKUS	2.000.000	Veröffentlichungskandidat	Abschließendes Review und Bereitstellung Veröffentlichungskandidat
11.10.2022	Christian Fuhrhop	Fraunhofer FOKUS	2.001.000	Kommentierbarer Arbeitsentwurf	Initiale APIs für Sichtfreigaben, /freigaben und /sichtbar
11.10.2022 – 12.10.2022	Christian Fuhrhop	Fraunhofer FOKUS	2.002.000	Kommentierbarer Arbeitsentwurf	APIs für Sichtfreigaben /freigaben, /sichtbar, /personenkontextfreigabe und /gruppenfreigabe
18.10.22	Christian Fuhrhop	Fraunhofer FOKUS	2.003.000	Kommentierbarer Arbeitsentwurf	Sortierindex hinzugefügt
24.10.2022 – 25.10.2022	Christian Fuhrhop	Fraunhofer FOKUS	2.004.000	Kommentierbarer Arbeitsentwurf	Freigaben-Endpunkte /personenkontextfreigabe und /gruppenfreigabe in einem Endpunkt zusammengefasst
28.10.2022	Christian Fuhrhop	Fraunhofer FOKUS	2.005.000	Kommentierbarer Arbeitsentwurf	Freigaben-Endpunkt umbenannt.
01.11.2022	Christian Fuhrhop	Fraunhofer FOKUS	2.006.000	Kommentierbarer Arbeitsentwurf	Referenzgruppen hinzugefügt

01.11.2022 – 02.11.2022	Christian Fuhrhop	Fraunhofer FOKUS	2.007.000	Kommentierbarer Arbeitsentwurf	Beschreibung des Ablaufs der Synchronisation mit einem primären Quellsystem hinzugefügt
14.11.2022	Christian Fuhrhop	Fraunhofer FOKUS	2.008.000	Kommentierbarer Arbeitsentwurf	Versionierung Beschreibung und API
15.11.2022	Christian Fuhrhop	Fraunhofer FOKUS	2.009.000	Kommentierbarer Arbeitsentwurf	Englischsprachiger Anhang
25.11.2022	Christian Fuhrhop	Fraunhofer FOKUS	2.010.000	Kommentierbarer Arbeitsentwurf	Beschreibung Funktion kombinierter Filter, Attribut "sichtfreigabe" bei Personenkontexten und Gruppen, Erweiterung Filter um Sichtfreigabe.
05.12.2022	Christopher Krauß, Christian Fuhrhop	Fraunhofer FOKUS	1.002.010.000	Kommentierbarer Arbeitsentwurf	Anpassung Versionierung; Versionsnummern der Spezifikation wurden um führende Stelle (Major-Release-Nummer) ergänzt. Releases mit der gleichen Major-Release-Nummer sind abwärtskompatibel.
06.12.2022	Christian Fuhrhop	Fraunhofer FOKUS	1.002.010.001	Kommentierbarer Arbeitsentwurf	Entfernung Einschluss-Filterparameter, Sichtfreigaben und Updates, Filtern von Attributen in Listenform
07.12.2022	Christian Fuhrhop	Fraunhofer FOKUS	1.002.011.000	Kommentierbarer Arbeitsentwurf	Attribut Stammorganisation hinzugefügt  Erweiterte Fehlermeldung bei Löschung von Personen und Personenkontexten. Ablaufbeschreibung Schuljahreswechsel
09.12.2022	Christian Fuhrhop	Fraunhofer FOKUS	1.002.012.000	Kommentierbarer Arbeitsentwurf	Mandant aus Gruppenzugehörigkeit entfernt.  Definition von Stammorganisation angepasst. Filter dürfen jeweils nur einmal benutzt werden.
13.12.2022	Christian Fuhrhop	Fraunhofer FOKUS	1.002.013.000	Kommentierbarer Arbeitsentwurf	Erste Version Löschkonzept. Beschreibung Vorgang Quellsystem, Beschreibung Dienste und sekundäre Systeme, API Personenkontexte-info.
19.12.2022	Christian Fuhrhop	Fraunhofer FOKUS	1.002.014.000	Kommentierbarer Arbeitsentwurf	Umbenennung API personenkontexte-info nach personen-info, Umstrukturierung Lehrfach in Fach, Thema und

					Beschreibung, Anpassung Filterbeschreibung
23.12.2022	Christian Fuhrhop	Fraunhofer FOKUS	1.002.015.000	Kommentierbarer Arbeitsentwurf	Bildungsziel statt Bildungsgang in Gruppen, Aufsplittung Attribut Löszeitpunkt in Attribute Löschung und Zeitpunkt Attribute von Fächer von Fach und Thema auf Kennung und Text geändert
27.12.2022	Christian Fuhrhop	Fraunhofer FOKUS	1.002.016.000	Kommentierbarer Arbeitsentwurf	Änderung personen-info API in Liste aktiver Personenkontexte statt gelöschter Personenkontexte
12.01.2023	Christian Fuhrhop	Fraunhofer FOKUS	1.002.017.000	Kommentierbarer Arbeitsentwurf	Thema und Beschreibung zu Gruppenattributen hinzugefügt. Attribut Beschreibung aus „Fach“ entfernt, Ergänzung der Ablaufbeschreibung Synchronisation um Gruppen und Gruppenzugehörigkeiten
03.02.2023	Christian Fuhrhop	Fraunhofer FOKUS	1.002.018.000	Kommentierbarer Arbeitsentwurf	Zeitformatanpassung Löszeitpunkt
16.02.2023	Christian Fuhrhop	Fraunhofer FOKUS	1.002.019.000	Kommentierbarer Arbeitsentwurf	Anpassung Kontext-ID von KTID auf ID, außer bei Gruppenzugehörigkeit
23.02.2023	Christian Fuhrhop	Fraunhofer FOKUS	1.002.020.000	Kommentierbarer Arbeitsentwurf	Abgleich Fehlermeldungen, Ersetzen von „SANIS“ durch „SchulConneX“, Ersetzen „Anmeldeserver“ durch „SchulConneX-Service“, Abarbeiten Kommentare, Initiale maximale Textlängen
24.02.2023	Christian Fuhrhop	Fraunhofer FOKUS	1.002.021.000	Kommentierbarer Arbeitsentwurf	Verbesserungen der Beschreibungen bei Textlängen, Lernperioden und Sortierindex, Abarbeitung weiterer Kommentare
28.02.2023	Christopher Krauß	Fraunhofer FOKUS	1.003.000.000	Veröffentlichungskandidat	Abschließendes Review und Bereitstellung Veröffentlichungskandidat
04.04.2023	Christian Fuhrhop	Fraunhofer FOKUS	1.003.001.000	Veröffentlichungskandidat	Einarbeitung von Rechtschreibkorrekturen und initialen Kommentaren
06.04.2023	Christian Fuhrhop	Fraunhofer FOKUS	1.003.002.000	Veröffentlichungskandidat	Lizenzhinweis auf CC BY-ND 2.0 DE, Einheitliche Nutzung von "SchulConneX-Service" und Aufnahme ins Glossar Verlagerung der nicht normativen "Best Practices" in eigenes Kapitel, Einpflegen Kommentare HKM



14.04.2023	Christopher Krauß	Fraunhofer FOKUS	1.003.003.000	Veröffentlichte Spezifikation	Abschließendes Review und Bereitstellung als Veröffentlichte Spezifikation
11.05.2023	Christian Fuhrhop	Fraunhofer FOKUS	1.004.001.000	Kommentierbarer Arbeitsentwurf	Wiederherstellung von für 1.3 gelöschten Teilen, insbesondere APIs /codelisten und /versionen.  Erste Version von Kontakt-Informationen für Personen, Kontexte und Organisationen
12.05.2023	Christian Fuhrhop	Fraunhofer FOKUS	1.004.002.000	Kommentierbarer Arbeitsentwurf	Filter für API <code>personen-info</code> zum Umfang der Ausgaben und zur Beschränkung auf einzelne Personen, Kontexte, Gruppen oder Organisationen
15.05.2023	Christian Fuhrhop	Fraunhofer FOKUS	1.004.003.000	Kommentierbarer Arbeitsentwurf	Erweiterte Beispiele der Filterfunktionen für API <code>personen-info</code>
19.05.2023	Christian Fuhrhop	Fraunhofer FOKUS	1.004.004.000	Kommentierbarer Arbeitsentwurf	Anpassung und Vereinfachung Datenmodell Kontakt. Erstellung und Einfügung der graphischen Darstellung des Datenmodells
24.05.2023	Christian Fuhrhop	Fraunhofer FOKUS	1.004.005.000	Kommentierbarer Arbeitsentwurf	Einfügung der graphischen Darstellung des vereinfachten Datenmodells für Dienste
02.06.2023	Christian Fuhrhop	Fraunhofer FOKUS	1.004.006.000	Kommentierbarer Arbeitsentwurf	Erweiterung der Code-Listen Jahrgangsstufe und Bildungsziel
07.06.2023	Christian Fuhrhop	Fraunhofer FOKUS	1.004.007.000	Kommentierbarer Arbeitsentwurf	Umbenennung Filterattribute und Filteroptionen bei API <code>personen-info</code> . Neues Attribut <code>änderungs_zeit</code> , Fehlermeldung doppelter Filter, Anpassung Kontakt zu Kommunikation, Attribut Namenspräfix entfernt, Mandant der Gruppenzugehörigkeit hinzugefügt
08.06.2023	Christian Fuhrhop	Fraunhofer FOKUS	1.004.008.000	Kommentierbarer Arbeitsentwurf	Anpassung Feldlängen
09.06.2023	Christian Fuhrhop	Fraunhofer FOKUS	1.004.009.000	Kommentierbarer Arbeitsentwurf	Mehrere Gruppenzugehörigkeiten bei <code>person-info</code>
12.06.2023	Christian Fuhrhop	Fraunhofer FOKUS	1.004.010.000	Kommentierbarer Arbeitsentwurf	Erweiterung von Personenkontext (Dienstanbieter) und <code>person-info</code> um Liste der PIDs weiterer Gruppenmitglieder

19.06.2023	Christian Fuhrhop	Fraunhofer FOKUS	1.004.011.000	Kommentierbarer Arbeitsentwurf	Beschreibung Best Practice für die Änderung der Rolle in einem Personenkontext, Anpassung Gruppenzugehörigkeit und Beispiel-JSON in person-info
20.06.2023	Christian Fuhrhop	Fraunhofer FOKUS	1.004.012.000	Kommentierbarer Arbeitsentwurf	Initiale Beschreibung OIDC Claims Nutzung und Attribut/Claim-Namen
21.06.2023	Christian Fuhrhop	Fraunhofer FOKUS	1.004.013.000	Kommentierbarer Arbeitsentwurf	Detaillierung der OIDC Claims Nutzung
26.06.2023	Christian Fuhrhop	Fraunhofer FOKUS	1.004.014.000	Kommentierbarer Arbeitsentwurf	Reduktion der OIDC Claims auf Kernattribut, Anpassung URN Format, Beispiele zum Scoping
27.06.2023	Christian Fuhrhop	Fraunhofer FOKUS	1.004.015.000	Kommentierbarer Arbeitsentwurf	Scoping für OIDC Claims in separates Kapitel ausgelagert, optionale Nutzung herausgestellt
27.06.2023	Christian Fuhrhop	Fraunhofer FOKUS	1.004.015.000	Kommentierbarer Arbeitsentwurf	Vereinfachung Scope-Nutzung bei OIDC Claims, Sonstige Gruppenzugehörige bei person-info
07.07.2023	Christian Fuhrhop	Fraunhofer FOKUS	1.004.016.000	Kommentierbarer Arbeitsentwurf	Erste Teilversion Beziehungen und Bezugspersonen
11.07.2023	Christian Fuhrhop	Fraunhofer FOKUS	1.004.017.000	Kommentierbarer Arbeitsentwurf	Erweiterte Version Beziehungen und Bezugspersonen
19.07.2023	Christian Fuhrhop	Fraunhofer FOKUS	1.004.018.000	Kommentierbarer Arbeitsentwurf	Erste Version Datentyp Anschrift (für Organisationen)
21.07.2023	Christian Fuhrhop	Fraunhofer FOKUS	1.004.019.000	Kommentierbarer Arbeitsentwurf	Angepasste Version Datentyp Anschrift (für Organisationen)
25.07.23 26.07.23 28.07.23	Christian Fuhrhop	Fraunhofer FOKUS	1.004.020.000	Kommentierbarer Arbeitsentwurf	Anpassung der Handhabung von Beziehungen an Muster Gruppenzugehörigkeiten, Nachtrag von Datenmodellen im englischen Teil
14.08.23	Christian Fuhrhop	Fraunhofer FOKUS	1.004.021.000	Kommentierbarer Arbeitsentwurf	Überarbeitung Bezugskontexte, eine Rolle pro Beziehung, Beziehungen sind 'Haben-' Beziehungen, Beispiele, API Umbenennung
15.08.23	Christian Fuhrhop	Fraunhofer FOKUS	1.004.022.000	Kommentierbarer Arbeitsentwurf	Filterfunktionen zur Anzeige von "Ist-"Beziehungen, Regeln zur Codelisten-Erweiterung
21.08.23 22.08.23	Christian Fuhrhop	Fraunhofer FOKUS	1.004.023.000	Kommentierbarer Arbeitsentwurf	Einarbeitung von diversen Kommentaren: Gruppenzugehörigkeiten, Adresszeile, OIDC Claims, Personen-Info, Beziehungen

25.08.23	Christian Fuhrhop	Fraunhofer FOKUS	1.004.024.000	Kommentierbarer Arbeitsentwurf	Fehlerhaftes Attribut 'Mandant' aus Beispiel Person-Info entfernt, Update-Funktion aus Beziehungen entfernt, eigene PKID beim Read hinzugefügt
08.09.23	Christian Fuhrhop	Fraunhofer FOKUS	1.004.025.000	Kommentierbarer Arbeitsentwurf	Stammorganisation zum Datenmodell für Dienstanbieter hinzugefügt
20.09.23	Christian Fuhrhop	Fraunhofer FOKUS	1.004.026.000	Kommentierbarer Arbeitsentwurf	Stammorganisation im Dienstanbieter-Datenmodell erweitert
09.10.23	Christian Fuhrhop	Fraunhofer FOKUS	1.004.027.000	Kommentierbarer Arbeitsentwurf	Beziehungen zwischen Organisationen
12.10.23	Christian Fuhrhop	Fraunhofer FOKUS	1.004.028.000	Kommentierbarer Arbeitsentwurf	Beziehungen in API für Dienste in person-info
18.10.23	Christian Fuhrhop	Fraunhofer FOKUS	1.004.029.000	Kommentierbarer Arbeitsentwurf	Verdeutlichung hat_als und ist_von Beziehungen, explizite Beschreibung in den Codelisten
23.10.23	Christian Fuhrhop	Fraunhofer FOKUS	1.004.030.000	Kommentierbarer Arbeitsentwurf	Erreichbarkeit auf E-Mail reduziert, RegEx hinzugefügt
24.10.23	Christian Fuhrhop	Fraunhofer FOKUS	1.004.031.000	Kommentierbarer Arbeitsentwurf	Beziehungen und Filter zu personen-info API hinzugefügt
03.11.23	Christian Fuhrhop	Fraunhofer FOKUS	1.004.032.000	Kommentierbarer Arbeitsentwurf	Einarbeitung Kommentare
10.11.23	Christian Fuhrhop	Fraunhofer FOKUS	1.004.033.000	Kommentierbarer Arbeitsentwurf	Hinzufügung von hat_als_ktid für Beziehungen, Kommunikationskanal für Dienste hinzugefügt
15.11.23	Christian Fuhrhop	Fraunhofer FOKUS	1.004.034.000	Kommentierbarer Arbeitsentwurf	Eigene API für Erreichbarkeit
17.11.23	Christian Fuhrhop	Fraunhofer FOKUS	1.004.035.000	Kommentierbarer Arbeitsentwurf	Überarbeitungen Beziehungen und Erreichbarkeit. Anpassung Fehler 401/18
20.11.23	Christian Fuhrhop	Fraunhofer FOKUS	1.004.036.000	Kommentierbarer Arbeitsentwurf	Code-Liste Erreichbarkeit in Erreichbarkeitstyp umbenannt. Keine direkte Änderbarkeit mehr über Person oder Personenkontext
24.11.23	Christian Fuhrhop	Fraunhofer FOKUS	1.004.037.000	Kommentierbarer Arbeitsentwurf	Spezifische Fehlermeldung zur Erreichbarkeit beim Create
29.11.23	Christian Fuhrhop	Fraunhofer FOKUS	1.004.038.000	Kommentierbarer Arbeitsentwurf	Integration Kommentare Personen-info, Attribut <code>änderungs_zeit</code> entfernt
01.12.23	Christian Fuhrhop	Fraunhofer FOKUS	1.004.039.000	Kommentierbarer Arbeitsentwurf	Separates Gruppenmodell für Dienste

08.12.23	Christian Fuhrhop	Fraunhofer FOKUS	1.004.040.000	Kommentierbarer Arbeitsentwurf	Angepasste Grafiken der Datenmodelle
15.12.23	Christian Fuhrhop	Fraunhofer FOKUS	1.004.041.000	Kommentierbarer Arbeitsentwurf	Anpassung CRUD Tabellen, diverse Umformulierungen, Einarbeitung Inhouse Review-Kommentare
11.1.24	Christian Fuhrhop	Fraunhofer FOKUS	1.004.042.000	Veröffentlichungskandidat	Erstellung Veröffentlichungskandidat