

Schnittstellenspezifikation

SchulConneX

Beschreibung der Schnittstellen von SchulConneX für Dienste und Quellsysteme

Version 1.003.000.000 (Veröffentlichungskandidat)

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

Stand vom 28.02.2023

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

Alle Teile dieses Dokuments sind urheberrechtlich geschützt und ausschließlich zum Gebrauch innerhalb des Projekts bestimmt. Dieses Dokument oder Teile davon dürfen Dritten nicht zugänglich gemacht werden und es ist dafür Sorge zu tragen, dass Dritte keinen Zugriff auf dieses und die Inhalte desselben erhalten. Über die Inhalte dieses Dokuments ist gegenüber Dritten Stillschweigen zu bewahren. Ausnahmen sind nur mit ausdrücklicher Erlaubnis des Verantwortlichen erlaubt.

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	14
2	VORBEMERKUNG	14
3	EINLEITUNG	14
4	ALLGEMEINES	15
4.1	Kernanwendungsfälle	15
4.2	Nutzung der Schnittstelle durch Dienst anbietende	15
4.3	Nutzung der Schnittstellen durch Quellsysteme	16
4.4	Ablaufbeschreibung: Synchronisation mit dem primären Quellsystem	16
4.4.1	Konfiguration	16
4.4.2	Synchronisation	16
4.5	Ablauf des Schuljahreswechsels für Quellsysteme	22
4.5.1	Besonderheiten bei der Aktualisierung der Personen und Personenkontexte	23
4.5.2	Besonderheiten beim Aktualisieren von Gruppen	23
4.6	Ablauf Löschen von Personenkontexten und Personen	26
4.6.1	Löschen von Personenkontexten und Personen aus Sicht des primären Quellsystems	26
4.6.2	Information von Diensten über die Löschung von Personenkontexten	27
5	DATENMODELL DER SCHNITTSTELLE	27
5.1	Datumsformat	28
5.2	Organisation	28
5.3	Person	30
5.4	Personenkontext	36
5.5	Personendatensatz	39
5.6	Gruppe	41
5.7	Gruppenzugehörigkeit	46
5.8	Gruppendatensatz	47
5.9	Lernperiode	49

6	ANGEPASSTES DATENMODELL FÜR DIENSTANBIETER-SCHNITTSTELLE	50
6.1	Angepasstes Datenmodell Person für Dienstanbieter	51
6.2	Angepasstes Datenmodell Personenkontext für Dienstanbieter	52
6.3	Angepasstes Datenmodell Organisation für Dienstanbieter	54
7	SCHNITTSTELLEDEFINITION	56
7.1	Allgemeine Festlegungen	56
7.1.1	Standard-Anfragen (Default Requests)	56
7.1.2	Standard-Rückgaben (Default Responses)	56
7.2	Erfolgreiche Ausführung	57
7.3	Daten unverändert	57
7.4	Fehlerbehandlung	57
7.5	Validierung der Anfrage	62
8	SCHNITTSTELLEN FÜR QUELLSYSTEME UND DIENSTE	63
8.1	API Versionierung	63
8.2	Registrierung	63
8.3	API Autorisierung für Quellsysteme	64
8.4	Autorisierung und Authentifikation bei Dienstanbietern	64
8.4.1	Access Token	66
8.4.2	ID Token	68
8.5	Vorbedingungen der Schnittstelle	69
9	API-ENDPUNKTE FÜR DIENSTANBIETER	71
9.1	Schnittstellenendpunkte für Personen /person-info	71
9.1.1	Schnittstelle „person-info“	71
9.2	Schnittstellenendpunkte für Personen /personen-info	76
9.2.1	Schnittstelle „personen-info“	76
10	API-ENDPUNKTE FÜR QUELLSYSTEME	79
10.1	Schnittstellen für Personen	79
10.1.1	[/personen] Personendatensätze erstellen und auflisten	79
10.1.2	[/personen/{id}] Individuelle Personendatensätze lesen, ändern und löschen	89
10.1.3	[/personen/{id}/personenkontexte] Personenkontexte lesen und erstellen	99
10.1.4	[/personenkontexte] Personenkontexte der eigenen Organisation lesen	107

10.1.5	<code>[/personenkontexte/{id}]</code>	Individuelle Personenkontexte lesen, ändern und löschen	113
10.2	Schnittstellen für Organisationen		123
10.2.1	<code>[/organisationen]</code>	Organisationen auflisten	123
10.2.2	<code>[/organisationen/{id}]</code>	Lesen einer individuellen Organisation	127
10.2.3	<code>[/organisation-info]</code>	Lesen der eigenen Organisation	130
10.3	Schnittstellen für Gruppen		133
10.3.1	<code>[/gruppen]</code>	Gruppen erstellen und lesen	133
10.3.2	<code>[/gruppen/{id}]</code>	Lesen, ändern und löschen einer individuellen Gruppe	144
10.3.3	<code>[/gruppen/{gruppe.id}/gruppenzugehoerigkeiten]</code>	Erstellen und Lesen von zu einer Gruppe gehörenden Gruppenzugehörigkeiten	154
10.3.4	<code>[/gruppenzugehoerigkeiten]</code>	Gruppenzugehörigkeiten der eigenen Organisation lesen	160
10.3.5	<code>[/gruppenzugehoerigkeiten/{gruppenzugehoerigkeit.id}]</code>	Lesen, ändern und löschen einer individuellen Gruppenzugehörigkeit	164
11	ALLGEMEINE API-ENDPUNKTE		172
11.1	Schnittstellen für Codelisten		172
11.1.1	<code>[/codelisten]</code>	Auflistung verfügbarer Codelisten	172
11.1.2	<code>[/codelisten/{codeliste}]</code>	Auflistung des Inhalts einer Codeliste	173
11.2	Schnittstellen für die Versionsverwaltung		174
11.2.1	<code>[/versionen]</code>	Auflistung verfügbarer Codelisten	174
12	CODELISTEN		176
12.1	Personenstatus		176
12.2	Geschlecht		176
12.3	Rolle		177
12.4	Vertrauensstufe		177
12.5	Organisationstyp		178
12.6	Jahrgangsstufe		178
12.7	Boolean		179
12.8	Trägerschaft		179
12.9	Lokalisierung		179
12.10	Gruppenbereich		180
12.11	Gruppendifferenzierung		180
12.12	Gruppenoption		180

12.13	Gruppentyp	181
12.14	Gruppenrolle	181
12.15	Lernperiode	181
12.16	Lernperiodentyp	182
12.17	Fächerkanon	182
12.18	Bildungsgang	184
12.19	Bildungsziel	185
13	GLOSSAR	186
14	ENGLISH API NOTES (ENG.)	188
14.1	Data Models (eng.)	188
14.1.1	Format of dates (eng.)	188
14.1.2	Organisation (eng.)	188
14.1.3	Person (eng.)	189
14.1.4	Personenkontext (eng.)	191
14.1.5	Personendatensatz (eng.)	192
14.1.6	Gruppe (eng.)	192
14.1.7	Gruppenzugehörigkeit (eng.)	194
14.1.8	Gruppendatensatz (eng.)	195
14.1.9	Freigabe (eng.)	195
14.1.10	Lernperiode (eng.)	196
14.2	Codelists (eng.)	196
15	ANLAGE: DOKUMENTHISTORIE	199

Abbildungsverzeichnis

Abbildung 1: Sequenzdiagramm für die Autorisierung und anschließende Beispielabfrage von Personendaten im Sicherheitskontext einer angemeldeten nutzenden Person 65

Datenmodelle

Datenmodell 1: Datenmodell einer Organisation	29
Datenmodell 2: Datenmodell einer Person	34
Datenmodell 3: Datenmodell eines Personenkontexts	38
Datenmodell 4: Datenmodell des Personendatensatzes	39
Datenmodell 5: Datenmodell der Gruppe	45
Datenmodell 6: Datenmodell der Gruppenzugehörigkeit	47
Datenmodell 7: Datenmodell des Gruppendatensatzes	47
Datenmodell : Datenmodell der Lernperiode	49
Datenmodell : Datenmodell Person für Dienstanbieter	52
Datenmodell : Datenmodell Personenkontext für Dienstanbieter	54
Datenmodell : Datenmodell Organisation für Dienstanbieter	55
Datenmodell : Access Token	67
Datenmodell : ID Token	69

Fehlercodelisten

Fehlerliste 1: Fehler 400 Bad Request	59
Fehlerliste 2: Fehler 401 Unauthorized	60
Fehlerliste 3: Fehler 403 Forbidden	61
Fehlerliste 4: Fehler 404 Not found	61
Fehlerliste 5: Fehler 405 Method not allowed	61
Fehlerliste 6: Fehler 409 Conflict	61
Fehlerliste 7: Fehler 500 Interner Serverfehler	62

Tabellenverzeichnis

Tabelle 1: Ablaufbeschreibung für den Abgleich der Datensätze	19
Tabelle 2: Ablaufbeschreibung für das Aktualisieren und Anlegen der Datensätze	19
Tabelle 3: Ablaufbeschreibung für den Abgleich der Datensätze (Gruppen und Gruppenzugehörigkeiten)	21
Tabelle 4: Ablaufbeschreibung für das Aktualisieren und Anlegen der Datensätze (Gruppen und Gruppenzugehörigkeiten)	22
Tabelle 5: Besonderheiten beim Aktualisieren fortgeführter Gruppen	24
Tabelle 6: Besonderheiten beim Aktualisieren beendeter Gruppen	25
Tabelle 7: Zugelassene Operationen auf dem Endpunkt „person-info“	71
Tabelle 8: Zugelassene Operationen auf dem Endpunkt „personen-info“	76
Tabelle 9: Zugelassene Operationen auf dem Endpunkt /v1/personen	79
Tabelle 10: Zugelassene Filter auf dem Endpunkt /v1/personen	80
Tabelle 11: Liste von Attributen, welche durch ein Quellsystem oder Dienst über den Endpunkt /v1/personen auf HTTP-POST nicht gesetzt werden können.	81
Tabelle 12: Zugelassene Operationen auf dem Endpunkt /v1/personen/{personendatensatz.person.id}	89
Tabelle 13: Liste von Attributen, welche durch ein Quellsystem oder Dienst über den Endpunkt /v1/personen/{personendatensatz.person.id} auf HTTP-PUT nicht verändert werden können.	92
Tabelle 14: Zugelassene Operationen auf dem Endpunkt /v1/personen/{personendatensatz.person.id}/personenkontexte	99
Tabelle 15: Zugelassene Filter auf dem Endpunkt /v1/personen/{personendatensatz.person.id}/personenkontexte	100
Tabelle 16: 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.	102
Tabelle 17: Zugelassene Operationen auf dem Endpunkt /v1/personenkontexte	107
Tabelle 18: Zugelassene Filter auf dem Endpunkt /v1/personenkontexte	108
Tabelle 19: Zugelassene Operationen auf dem Endpunkt /v1/personenkontexte/{personendatensatz.personenkontext.id}	113
Tabelle 20: Liste von Attributen, welche durch ein Quellsystem oder Dienst über den Endpunkt /v1/personenkontexte/{personendatensatz.personenkontext.id} auf HTTP-PUT nicht verändert werden können.	118
Tabelle 21: Zugelassene Operationen auf dem Endpunkt /v1/organisationen	123

Tabelle 22: Zugelassene Filter auf dem Endpunkt /v1/organisationen	124
Tabelle 23: Zugelassene Operationen auf dem Endpunkt /v1/organisationen/{organisation.id}	127
Tabelle 24: Zugelassene Operationen auf dem Endpunkt /v1/organisation-info	130
Tabelle 25: Zugelassene Operationen auf dem Endpunkt /v1/gruppen	133
Tabelle 26: Zugelassene Filter auf dem Endpunkt /v1/gruppen	138
Tabelle 27: Zugelassene Operationen auf dem Endpunkt /v1/gruppen/{gruppe.id}	144
Tabelle 28: Liste von Attributen, welche durch ein Quellsystem oder Dienst über den Endpunkt /v1/gruppen/{gruppe.id} auf HTTP-PUT nicht verändert werden können.	148
Tabelle 29: Zugelassene Operationen auf dem Endpunkt /v1/gruppen/{gruppe.id}/gruppenzugehoerigkeiten	154
Tabelle 30: Liste von Attributen, welche durch ein Quellsystem oder Dienst über den Endpunkt /v1/gruppen/{gruppe.id} /gruppenzugehoerigkeiten auf HTTP-POST nicht gesetzt werden können.	155
Tabelle 31: Zugelassene Filter auf dem Endpunkt /v1/gruppen/{gruppe.id}/gruppenzugehoerigkeiten /v1/gruppen	158
Tabelle 32: Zugelassene Operationen auf dem Endpunkt /v1/gruppenzugehoerigkeiten	160
Tabelle 33: Zugelassene Filter auf dem Endpunkt /v1/gruppenzugehoerigkeiten	161
Tabelle 34: Zugelassene Operationen auf dem Endpunkt /v1/gruppenzugehoerigkeiten/{gruppenzugehoerigkeit.id}	164
Tabelle 35: Liste von Attributen, welche durch ein Quellsystem oder Dienst über den Endpunkt /v1/gruppenzugehoerigkeit/{gruppenzugehoerigkeit.id} auf HTTP-PUT nicht verändert werden können.	168
Tabelle : Zugelassene Operationen auf dem Endpunkt /v1/codelisten	172
Tabelle : Zugelassene Operationen auf dem Endpunkt /v1/codelisten/{codeliste}	173
Tabelle : Zugelassene Operationen auf dem Endpunkt /v1/versionen	174

Quellcodes

Quellcode 1: Beispiel JSON für eine Organisation	30
Quellcode 2: Beispiel JSON einer Person	35
Quellcode 3: Beispiel JSON eines Personenkontexts	39
Quellcode 4: Beispiel JSON eines Personendatensatzes	41
Quellcode 5: Beispiel JSON eines Gruppendatensatzes	49
Quellcode : Beispiel des Error Payloads	57

Quellcode : URL-Struktur der Schnittstelle	63
Quellcode : Beispiel der URL eines Schnittstellenendpunkts	63
Quellcode : Beispiel eines Authorization Headers	69
Quellcode : Beispiel eines Accept Headers	70
Quellcode : Beispiel des Schnittstellen-Endpunkts für person-info	71
Quellcode : JSON-Struktur der Rückgabe zu person-info	72
Quellcode : Beispiel einer JSON-Rückgabe zu person-info	74
Quellcode : Beispiel des Schnittstellen-Endpunkts für personen-info	76
Quellcode : JSON-Struktur der Rückgabe zu personen-info	77
Quellcode : Beispiel einer JSON-Rückgabe zu personen-info	78
Quellcode : URL-Struktur des Endpunkts personen	79
Quellcode : JSON-Struktur eines CREATE-Requests an /v1/personen	82
Quellcode : Beispiel Request zu /personen auf HTTP-POST	83
Quellcode : JSON-Struktur einer CREATE-Response an /v1/personen	83
Quellcode : Beispiel Response zu /personen auf HTTP-POST	84
Quellcode : JSON-Struktur eines READ-Requests an /v1/personen	85
Quellcode : Beispiel Response zu /personen auf HTTP-GET	87
Quellcode : Beispiel Response zu /personen auf HTTP-GET mit den Filter-parametern (query-parameter) „?referrer=123“	88
<i>Quellcode : URL-Struktur des Endpunkts personen/{personendatensatz.person.id}</i>	89
Quellcode : JSON-Struktur der Antwort bei READ auf /{personendatensatz.person.id}	90
Quellcode : Beispiel Response zu /personen/{personendatensatz.person.id} auf HTTP-GET	91
Quellcode : JSON-Struktur eines READ-Requests an /v1/personen/{personendatensatz.person.id}	93
Quellcode : Spezielle JSON-Struktur eines READ-Requests an /v1/personen/{personendatensatz.person.id}	94
Quellcode : Beispiel Request zu /personen/{personendatensatz.person.id} auf HTTP-PUT	94
Quellcode : Beispiel Request zu /personen/{personendatensatz.person.id} auf HTTP-PUT	95
Quellcode : Beispiel Response zu /personen/{personendatensatz.person.id} auf HTTP-PUT	96
Quellcode : JSON-Struktur eines DELETE-Requests an /v1/personen/{personendatensatz.person.id}	97
Quellcode : Beispiel Request zu /personen/{personendatensatz.person.id} auf HTTP-DELETE	97

<i>Quellcode : URL-Struktur des Endpunkts</i>	
<i>personen/{personendatensatz.person.id}/personenkontexte</i>	99
Quellcode : JSON-Struktur eines CREATE-Requests an <i>/v1/personen/{personendatensatz.person.id}/personenkontexte</i>	102
Quellcode : Beispiel Request zu <i>/personen/{personendatensatz.person.id}/personenkontexte</i> auf HTTP-POST	102
Quellcode : JSON-Struktur einer CREATE-Response an <i>/v1/personen/{personendatensatz.person.id}/personenkontexte</i>	103
Quellcode : Beispiel Response zu <i>/personen/{personendatensatz.person.id}/personenkontexte</i> auf HTTP-POST	103
Quellcode : JSON-Struktur einer READ-Response an <i>/v1/personen/{personendatensatz.person.id}/personenkontexte</i>	104
Quellcode : Beispiel Response zu <i>/personen/{personendatensatz.person.id}/personenkontexte</i> auf HTTP-GET	105
Quellcode : Beispiel Response zu <i>/personen/{personendatensatz.person.id}/personenkontexte</i> auf HTTP-GET mit den Filter- parametern (query-parameter) „?referrer=NI_56-78-90_123“	106
Quellcode 43: URL-Struktur des Endpunkts <i>personenkontexte</i>	107
Quellcode 44: JSON-Struktur einer READ-Response an <i>/v1/personenkontexte</i>	110
Quellcode 45: Beispiel Response zu <i>/personenkontexte</i> auf HTTP-GET	111
Quellcode 46: Beispiel Response zu <i>personenkontexte</i> auf HTTP-GET mit den Filter- parametern (query-parameter) „?referrer=NI_56789_123“	111
<i>Quellcode : URL-Struktur des Endpunkts</i>	
<i>personenkontexte/{personendatensatz.personenkontext.id}</i>	113
Quellcode : JSON-Struktur der Antwort bei READ auf <i>/v1/personenkontexte/{id}</i>	114
Quellcode : JSON-Struktur einer READ-Response an <i>/v1/personenkontexte/{id}</i>	115
Quellcode : Beispiel Response zu <i>/personenkontexte/{personendatensatz.personenkontext.id}</i> auf HTTP-GET	116
Quellcode : JSON-Struktur eines UPDATE-Requests an <i>/v1/personenkontexte/{id}</i>	118
Quellcode : Beispiel Request zu <i>/personenkontexte/{personendatensatz.personenkontext.id}</i> auf HTTP-PUT	119
Quellcode : JSON-Struktur einer UPDATE-Response an <i>/v1/personenkontexte/{id}</i>	119
Quellcode : Beispiel Response zu <i>/personenkontexte/{personendatensatz.personenkontext.id}</i> auf HTTP-PUT	120
Quellcode : JSON-Struktur eines DELETE-Requests an <i>/v1/personenkontexte/{id}</i>	121

Quellcode : Beispiel Request zu /personenkontexte/{personendatensatz.personenkontext.id} auf HTTP-DELETE	121
<i>Quellcode : URL-Struktur des Endpunkts organisationen</i>	123
Quellcode : JSON-Struktur einer READ-Response an /v1/organisationen	125
Quellcode : Beispiel Response zu /organisationen auf HTTP-GET	125
Quellcode : Beispiel Response zu /organisationen auf HTTP-GET mit den Filter-parametern „?kennung=NI_12-34-56&typ=SCHULE“	126
<i>Quellcode : URL-Struktur des Endpunkts organisationen/{organisation.id}</i>	127
Quellcode : JSON-Struktur einer READ-Response an /v1/organisationen/{id}	128
Quellcode : Beispiel Response zu /organisationen/{organisation.id} auf HTTP-GET	128
<i>Quellcode : URL-Struktur des Endpunkts organisation-info</i>	130
Quellcode : JSON-Struktur einer READ-Response an /v1/organisation-info	131
Quellcode : Beispiel Response zu /organisation-info auf HTTP-GET	131
Quellcode : URL-Struktur des Endpunkts Gruppen	133
Quellcode : JSON-Struktur eines CREATE-Requests an /v1/gruppen	135
Quellcode : JSON-Struktur einer CREATE-Response an /v1/gruppen	135
Quellcode : Beispiel Response zu /gruppen auf HTTP-POST	136
Quellcode : JSON-Struktur eines READ-Requests an /v1/personen	139
Quellcode : JSON-Struktur einer READ-Response an /v1/Gruppe	140
Quellcode : Beispiel Response zu /gruppen auf HTTP-GET	143
<i>Quellcode : URL-Struktur des Endpunkts gruppen/{gruppe.id}</i>	144
Quellcode : JSON-Struktur der Antwort bei READ auf /gruppen/{gruppe.id }	145
Quellcode : Beispiel Response zu /gruppen/{gruppe.id} auf HTTP-GET	146
Quellcode : JSON-Struktur eines UPDATE-Requests an /v1/gruppen/{gruppen.id}	150
Quellcode : JSON-Struktur einer UPDATE-Response an /v1/gruppen/{gruppe.id}	151
Quellcode : Beispiel Response zu /gruppen/{gruppe.id } auf HTTP-PUT	152
Quellcode : JSON-Struktur eines DELETE-Requests an /v1/gruppen/{gruppe.id}	153
Quellcode : Beispiel Request zu /v1/gruppen/{gruppe.id} auf HTTP-DELETE	153
<i>Quellcode : URL-Struktur des Endpunkts gruppen/{gruppe.id}/gruppenzugehoerigkeiten</i>	154
Quellcode : JSON-Struktur eines CREATE-Requests an /v1/gruppen/{gruppe.id }/gruppenzugehoerigkeiten	156
Quellcode : Beispiel Request zu /v1/gruppen/{gruppe.id }/ gruppenzugehoerigkeiten auf HTTP-POST	156

Quellcode : JSON-Struktur einer CREATE-Response an /v1/gruppen/{gruppe.id}/gruppenzugehoerigkeiten	156
Quellcode : Beispiel Response zu /v1/gruppen/{gruppe.id}/gruppenzugehoerigkeiten auf HTTP-POST	157
Quellcode : JSON-Struktur einer READ-Response an /v1/ gruppen/ {gruppe.id}/gruppenzugehoerigkeiten	159
Quellcode : Beispiel Response zu /gruppen/{gruppe.id}/gruppenzugehoerigkeiten auf HTTP- GET	159
Quellcode : URL-Struktur des Endpunkts gruppenzugehoerigkeiten	160
Quellcode : JSON-Struktur einer READ-Response an /v1/gruppenzugehoerigkeiten	162
Quellcode : Beispiel Response zu /gruppenzugehoerigkeiten auf HTTP-GET	162
<i>Quellcode : URL-Struktur des Endpunkts gruppenzugehoerigkeiten/{gruppenzugehoerigkeit.id}</i>	164
Quellcode : JSON-Struktur einer READ-Response an /v1/gruppenzugehoerigkeiten/{gruppenzugehoerigkeit.id}	166
Quellcode : Beispiel Response zu /gruppenzugehoerigkeiten/{gruppenzugehoerigkeit.id} auf HTTP-GET	167
Quellcode : JSON-Struktur eines UPDATE-Requests an /v1/gruppenzugehoerigkeiten/{gruppenzugehoerigkeit.id}	169
Quellcode : JSON-Struktur einer UPDATE-Response an /v1/gruppenzugehoerigkeiten/{gruppenzugehoerigkeit.id}	169
Quellcode : Beispiel Response zu /gruppenzugehoerigkeiten/{gruppenzugehoerigkeit.id } auf HTTP-PUT	170
Quellcode : JSON-Struktur eines DELETE-Requests an /v1/gruppenzugehoerigkeiten/{gruppenzugehoerigkeit.id}	171
Quellcode : Beispiel Request zu /v1/gruppenzugehoerigkeiten/{gruppenzugehoerigkeit.id} auf HTTP-DELETE	171
Quellcode : URL-Struktur des Endpunkts codelisten	172
Quellcode : JSON-Struktur eines READ-Requests an /v1/codelisten	173
Quellcode : URL-Struktur des Endpunkts codelisten/{codeliste}	173
Quellcode : JSON-Struktur eines READ-Requests an /v1/codelisten/geschlecht	174
Quellcode : URL-Struktur des Endpunkts versionen	174
Quellcode : JSON-Struktur eines READ-Requests an /v1/versionen	175

1 Stand der Bearbeitung

Dieses Dokument befindet sich aktuell im Bearbeitungsstand „Veröffentlichungskandidat“ in Version 1.

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

Die Schnittstellen dienen der Bereitstellung eines zentralen Identitätsdienstes für schulische Zwecke unter Berücksichtigung gängiger Standards und Technologien. In der Zukunft werden diese Schnittstellen um weitere Funktionen erweitert, welche dann auch die zentrale Bereitstellung von weiteren schulischen Daten aus unterschiedlichen Schulverwaltungssystemen für angeschlossene Dienste ermöglichen.

Dieses Dokument spezifiziert die Schnittstellen für SchulConneX. Die Definition und Bereitstellung dieser Schnittstellen (APIs) ist ein 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, mit dem Ziel, standardisierte Anmeldeverfahren für den Zugang zu digitalen Bildungsplattformen und -medien zu schaffen.

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

4 Allgemeines

4.1 Kernanwendungsfälle

Prinzipiell unterstützt die API zwei grundlegend unterschiedliche Anwendungsfälle: Zum einen die Nutzung als Anmeldeserver für Dienstanbieter, zum anderen als System, welches es Quellsystemen erlaubt, Daten zu synchronisieren.

Der Zugriff auf alle Endpunkte des REST-APIs des Anmeldeservice werden durch einen zentralen Authentifizierungs- und Autorisierungsserver gesichert. Ein Zugriff auf die Endpunkte ist nur mit einem gültigen, von diesem Server ausgestellten Access-Token möglich. 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 Dienstbietende

Alle, die ein Bildungsangebot nutzen, besitzen ein Konto bei einem Identity Provider. Will sich jemand an einem Dienst anmelden, so leitet diese Person die Anmeldung an den Anmeldeservice weiter. Der oder die Interessierte meldet sich dort an. Der Anmeldeservice stellt daraufhin dem Dienst die zu dessen Nutzung notwendigen Informationen über die nutzende Person bereit.

Dienste haben die Möglichkeit, das API des Anmeldeservices zu nutzen und darüber Daten des Anmeldeservices abzurufen. Bei den Zugriffen auf das REST-API können sie sich in unterschiedlichen Sicherheitskontexten bewegen:

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.

Aufruf im **Sicherheitskontext einer angemeldeten Person mit Zugriff auf alle Personenkontexte**, die für den Dienst freigegeben sind. Nur denen mit besonderem Vertrauensstatus kann es ermöglicht werden, auf die vollständigen Daten einer Person (alle Personenkontexte) zuzugreifen.

Aufruf im **Sicherheitskontext der Dienstbietenden 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 Anmeldeservice, ohne dass sich die nutzenden Personen aktiv anmelden müssen. Hierdurch ist es beispielsweise möglich, vollständige Informationen zu Gruppen und Gruppenzugehörigkeiten innerhalb von Organisationen bei einem Dienst abzubilden, bevor sich die Gruppenbeteiligten zum ersten Mal über den Anmeldeservice bei ihm anmelden. Diese Zugriffsart wird nur Diensten mit dem höchsten Vertrauensstatus angeboten.

4.3 Nutzung der Schnittstellen durch Quellsysteme

Quellsysteme haben die Möglichkeit, das Quellsysteme-API des Anmeldeservices zu nutzen und darüber Daten mit der Datenbank des Anmeldeservices zu synchronisieren.

4.4 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-Servers. Aus dem Vorschlag lassen sich die notwendigen Teile der Schnittstellenimplementierung auf Seiten des Quellsystems ableiten.

4.4.1 Konfiguration

Das Quellsystem muss sich für die Nutzung der REST API des SchulConneX-Servers authentifizieren. Dafür wird der Password Credentials Grant des OAuth2-Flows 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 Nutzerinnen und Nutzer 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.

Anforderungen an das Quellsystem

- Implementierung eines Unterpunkts im Konfigurationsmenü zur Eingabe und zum Abspeichern der Zugangsdaten
- Implementierung eines Rechts "Synchronisation mit SchulConneX-Server" und der Möglichkeit, dieses Recht an Rollen oder Benutzer oder Benutzerinnen zu vergeben.

4.4.2 Synchronisation

Der Synchronisationsvorgang besteht aus mehreren Schritten.

1. Auslösen des Synchronisationsvorgangs
2. Auswahl der zu synchronisierenden Gruppen
3. Abruf der Personen des Mandanten des Quellsystemes aus dem SchulConneX-Server
4. Abgleich der Datensätze
5. Aktualisieren und Anlegen der Datensätze
6. Importieren bestätigter Datensätze in das Quellsystem
7. Abruf der Gruppen des Mandanten des Quellsystems aus dem SchulConneX-Server
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 Quellsystemes aus dem SchulConneX-Server

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-Servers verwendet werden. In diesem Fall muss das Quellsystem die UUID aus dem SchulConneX-Server als Fremdschlüssel selbst speichern. Alternativ kann das Quellsystem seinen eigenen Schlüssel als Fremdschlüssel ("referrer") im SchulConneX-Server ablegen und den Abgleich darüber durchführen.

Falls beide Möglichkeiten genutzt werden, sollte das Quellsystem eine Strategie festlegen, wie bei Schlüsselkonflikten verfahren wird.

Fall	Beschreibung	Aktion
1	Der Datensatz im SchulConneX-Server kann einem Datensatz im Quellsystem über einen der beiden Schlüssel eindeutig zugeordnet werden.	Vormerken für automatische Aktualisierung
2	Sowohl im Datensatz im SchulConneX-Server 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-Server 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-Server 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, nachdem das Quellsystem bereits neue Personen im SchulConneX-Server hat.)	Vormerken für Import nach Import-Bestätigung
5	Der Datensatz im SchulConneX-Server 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-Server zugeordnet werden.	Vormerken zum automatischen Anlegen eines neuen Datensatzes im SchulConneX-Server
---	--	---

Tabelle 1: Ablaufbeschreibung für den Abgleich der Datensätze

Anforderungen and 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 Fälle 4 und 5 (Import)

Die Möglichkeit, unbekannte Datensätze aus dem SchulConneX-Server 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 2: 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-Server 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-Server

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-Server 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-Server kann einem Datensatz im Quellsystem über einen der beiden Schlüssel eindeutig zugeordnet werden.	Vormerken für automatische Aktualisierung	
2	Der Datensatz aus dem SchulConneX-Server hat einen Fremdschlüssel (referrer) des Quellsystems, ist aber anhand des Schlüssels im Quellsystem nicht auffindbar.	Vormerken für Import nach Import-Bestätigung	Dieser Fall kann z. B. auftreten, wenn im Quellsystem ein Backup wiedereingespielt wurde, nachdem das Quellsystem bereits neue Gruppen im SchulConneX-Server angelegt hat.
3	Der Datensatz aus dem SchulConneX-	Vormerken für Import nach	Dieser Fall tritt auf, wenn Gruppen in der Administrationsoberfläche des SchulConneX-Servers manuell oder durch

	Server kann einem Datensatz im Quellsystem nicht zugeordnet werden.	Import-Bestätigung	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-Server zugeordnet werden.	Vormerken zum automatischen Anlegen eines neuen Gruppen-Datensatzes im SchulConneX-Server	

Tabelle 3: 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
Aktualisieren	PUT /gruppen/{uuid} PUT /gruppenzugehoerigkeiten/{uuid}	Die Revision muss als Attribut mitgegeben werden
Anlegen	POST /gruppen/ POST /gruppen/{uuid}/ gruppenzugehoerigkeiten/	Jede Gruppenzugehörigkeit muss einzeln angelegt werden.
Löschen	DELETE /gruppen/ DELETE /gruppenzugehoerigkeiten/{uuid}	Jede Gruppenzugehörigkeit muss einzeln gelöscht werden.

Tabelle 4: 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 4.5 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 Nutzer oder die Nutzerin dem Import zugestimmt hat und die Attribute im SchulConneX-Server zum Anlegen eines neuen Datensatzes im QS 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.

4.5 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-Server 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-Servers 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.

4.5.1 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-Server 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 Schüler) muss meist das Feld "Personenkontext.Jahrgangsstufe" aktualisiert werden.

4.5.2 Besonderheiten beim Aktualisieren von Gruppen

Gruppen, die über den Schuljahreswechsel hinweg fortbestehen, sollen auf keinen Fall gelöscht werden. Sie sollen im SchulConneX-Server 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-Server 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 5: 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-Server
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-Server, 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 6: 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 soll, 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 Quellsysteme 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-Server" 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.

4.6 Ablauf Löschen von Personenkontexten und Personen

Aus Sicht des Datenschutzes ist der Anmeldeservice ein Datenübermittler zwischen Schulverwaltung (Quellsystemen) und angebundenen Anwendungen (Dienste). Aus dieser Voraussetzung leiten sich bestimmten 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, die Unverzüglichkeit der Löschung und die Benachrichtigung anderer Organisationen, denen die Daten bereitgestellt wurden, über die Löschung der Daten. Diese Organisation sind dann aufgefordert, eventuelle Kopien der Daten ihrerseits unverzüglich zu löschen.

Der SchulConneX-Server 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.

4.6.1 Löschen von Personenkontexten und Personen aus Sicht des primären Quellsystems

Um einen Persondatensatz und die dazugehörenden 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 Löschezitpunktes gelöscht. Wurden Personenkontexte noch nicht von anderen Diensten genutzt, so können diese auch direkt mit dem HTTP DELETE Aufruf der API "personenkontexte/{id}" gelöscht werden. Wurden die Daten jedoch an eine andere Organisation weitergeleitet, ist die direkte Löschung nicht mehr möglich und die Löschung muss durch Setzen des Löschezitpunktes ausgelöst werden.

Dabei ist der Vorgang wie folgt:

- Es wird ein Löschezitpunkt für jeden Personenkontext der Person gesetzt.
- Dieser Löschezitpunkt darf nicht in der Vergangenheit liegen.
- Zum Löschezitpunkt wird der Personenkontext vom SchulConneX-Server gelöscht, die darin vorhandenen Daten sind nicht mehr direkt zugreifbar.
- Danach sind keine Personenkontexte mehr mit der Person verbunden und die Daten der Person können mit der API "personen/{id}" gelöscht werden.

4.6.2 Information von Diensten über die Löschung von Personenkontexten

Die DSGVO schreibt vor, dass Dienste, welchen personenbezogenen 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" bereit gestellt.

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 Löschezitpunkt. 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 Löschezitpunkts allein ist aber nicht ausreichend, um die Löschung vorzunehmen, da der Löschezitpunkt 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.

Sind zu einer Person alle lokal persistent gehaltenen Personenkontexte gelöscht, so sind auch die persistent gehaltenen Daten zur Person zu 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 werden 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. Diese umfassen die Attribute mit der Anzahl 1. Beispielsweise kann ein Datenmodell nur das Attribut id (identifizier) enthalten.

Bei Attributen vom Typ String (Zeichenkette) ist die maximale Länge der Zeichenkette standardmäßig mit 256 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.

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.

5.2 Organisation

Eine Organisation stellt eine Schule oder sonstige relevante Organisation im Kontext des Schulischen Anmeldeservice 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
id	String (UTF-8)	1	ID der Organisation. Wird vom Anmeldeservice vergeben und ist eindeutig. Dieses Attribut ist unveränderbar (immutable).
kennung	String (UTF-8)	0..1	Die optionale Kennung (Identifikations-ID) einer "Organisation" muss innerhalb eines Organisationstyps eindeutig sein.
name	String (DIN 91379.B)	0..1	Offizieller Name einer Organisation
namensergaenzung	String (DIN 91379.B)	0..1	Ergänzender Name einer Organisation.
kuerzel	String (64) (DIN 91379.B)	0..1	Kurzname einer Organisation.
typ	String (Code)	0..1	Typ der Organisation. Referenz auf einen Code der Codeliste 12.5 Organisationstyp
traegerschaft	String (Code)	0..1	Art der Trägerschaft der Organisation. Referenz auf einen Code der Codeliste 12.8 Trägerschaft

Datenmodell 1: Datenmodell einer Organisation

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"
}
```

5.3 Person

Nachfolgend ist das Datenmodell einer Person dargestellt.

Attribut	Typ	Anz.	Bemerkung
id	String (UTF-8)	1	ID der Person. Wird vom Anmeldeservice vergeben ist eindeutig. Dieses Attribut ist unveränderbar (immutable).
referrer	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).
mandant	String (UTF-8)	1	ID des Mandanten, dem die Personen zugeordnet ist. Wird vom Anmeldeservice vergeben und ist eindeutig.
stammorganisation	String (UTF-8)	0..1	Personen können einer Organisation angehören, jedoch zeitweise an einer anderen Organisation tätig sein.
name	Strukturierung	1	

Attribut	Typ	Anz.	Bemerkung
name.familienname	String (DIN 91379.A)	1	Familienname der Person
name.vorname	String (DIN 91379.A)	1	Mehrere Vornamen werden durch Leerzeichen separiert.
name.initialenFamilienname	String (DIN 91379.A)	0..1	Initial oder Initialen des Familiennamens.
name.initialenVorname	String (DIN 91379.A)	0..1	Initial oder Initialen des Vornamens.
name.rufname	String (DIN 91379.A)	0..1	Bei mehreren Vornamen wird der täglich im Gebrauch genutzte Vorname geführt.
name.titel	String (DIN 91379.B)	0..1	Titel der Person (bspw. ein akademischer Grad, Dienst- und Amtsbezeichnungen oder militärische Ränge).
name.anrede	String (64/1024) (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 1024 Zeichen.
name.namenspräfix	String (64/1024) (DIN 91379.A)	0..n	Liste (Array) von Namenspräfixen.

Attribut	Typ	Anz.	Bemerkung
			Jeder einzelne Namenspräfix hat eine Maximallänge von 64 Zeichen, die Maximallänge für alle Namenspräfixe liegt bei 1024 Zeichen.
name.namenssuffix	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.
name.sortierindex	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 werden ab dem fünften Buchstaben des Nachnames sortiert werden soll. Ist kein Sortierindex angegeben, so ist der Nachname entsprechend dem ersten Buchstaben in Listen einzusortieren, äquivalent zum Sortierindex "0".
geburt	<i>Strukturierung</i>	0..1	

Attribut	Typ	Anz.	Bemerkung
geburt.datum	Datum (siehe Abschnitt 5.10)	0..1	Geburtsdatum einer Person
geburt.geburtsort	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.
geschlecht	String (Code)	0..1	Referenz auf einen Code der Codeliste 12.2 Geschlecht
lokalisierung	String (Code)	0..1	bevorzugte Lokalisierungseinstellung einer Person für Anwendungen gemäß RFC5646. Wenn nicht angegeben wird „de“ angenommen. Referenz auf einen Code der Codeliste 12.9 Lokalisierung.
vertrauensstufe	String (Code)	0..1	gibt an, wie stark die Personendaten vom erfassenden Mandanten verifiziert wurden. Referenz auf einen Code der Codeliste 12.4 Vertrauensstufe.
auskunftssperre	String (Code)	1	Auskunftssperre über eine Person. Dienste

Attribut	Typ	Anz.	Bemerkung
			<p>erhalten nur einen uneingeschränkten Datensatz, wenn der Wert "NEIN" ist. Sofern nicht explizit "JA" eingegeben wurde, ist der Default "NEIN".</p> <p>Referenz auf einen Code der Codeliste 12.7</p> <p>Boolean</p>
revision	String (UTF-8)	1	<p>Revision der Person. Wird vom Anmeldeservice mit der Erstellung des Datensatzes sowie Aktualisierung generiert. Dieser Wert kann nicht von Quellsystemen oder Diensten gesetzt werden.</p>

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 Information 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 „Auszubildender“ 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 zu 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 Anmeldeservice 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 Anmeldeservice 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.
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 12.612.6 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 an dem dieser Personenkontext gelöscht wird oder wurde. Das Format des Löschzeitpunktes 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 Anmeldeservice 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",
    "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
person	Person	1	Personen. (Siehe 5.3 Datenmodell Personen)
personenkontexte	Personenkontext	0..n	Liste (Array) von Personenkontexte. (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",
      "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 Anmeldeservice vergeben und ist eindeutig. Dieses Attribut ist unveränderbar (immutable).

mandant	String (UTF-8)	1	ID des Mandanten, dem die Gruppe zugeordnet ist. Wird vom Anmeldeservice 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 Mandant, 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, Arbeitsgemeinschaft...) 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 12.19 Bildungsziel.
jahrgangsstufen	String (Code)	0..n	Jahrgangsstufe (Jahrgangsstufe 1 bis Jahrgangsstufe 10) Referenz auf Code der Codeliste 12.1212.6 Jahrgangsstufe.
faecher	Strukturierung	0..n	Liste der in der Gruppe behandelte Fächer
faecher.code	String (Code)	0..1	In der Gruppe behandeltes Fach. Referenz auf Code der Codeliste 12.1212.17 Fächerkanon .
referenzgruppen	Strukturierung	0..n	<p>Array von Referenzgruppen zu dieser Gruppe.</p> <p>Bei Referenzgruppen handelt es sich um Gruppen, deren Mitglieder vollständig der oder teilweise 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</p>

			<p>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.id	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. <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>

			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.
laufzeit.von	Datum (siehe Abschnitt 5.1)	0..1	Anfang des Gültigkeitszeitraumes der Gruppe
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 Anmeldeservice 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
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
bis	Datum (siehe Abschnitt 5.1)	0..1	Ende der Gruppenzugehörigkeit
revision	String (UTF-8)	1	Revision der Gruppenzugehörigkeit. Wird vom Anmeldeservice mit der Erstellung des Datensatzes sowie Aktualisierung generiert. Dieser Wert kann nicht von

			Quellsystemen oder Diensten gesetzt werden.
--	--	--	---

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
gruppe	Gruppe	1	Gruppe. (Siehe 5.6 Gruppe)
gruppenzugehoerigkeiten	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"
},
"gruppzugehoerigkeiten": [{
    "id": "8722b058-8747-4646-8da0-d523a9f619b8",
    "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

6 Angepasstes Datenmodell für Dienstanbieter-Schnittstelle

Zur Anmeldung beim Dienstanbieter ist nicht die volle Komplexität erforderlich, die einem Quellsystem zur Verfügung stehen muss. Vorrangig werden nur solche Daten bereitgestellt, die zur Anmeldung von Nutzenden an einen Dienst notwendig sind.

So sind hier beispielsweise Namenszusätze, Geburtsort oder die von einem Quellsystem vergebene ID nicht von Relevanz.

Darüber hinaus wird zur Referenzierung der Daten einer spezifischen Person oder von Personenkontexten eine pseudonymisierte ID (pid) genutzt, welche nur aus Sicht der einzelnen Dienste konstant und eindeutig ist. Diese pseudonymisierte ID entspricht nicht der intern in der Datenhaltung des Anmelde-Servers 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 Angeboten, unter Umgehung des Anmelde-Servers, 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 Dienstanbietern zur Verfügung. Welche Daten von Anmeldeservice an den Dienst geliefert werden, wird durch individuelle Absprachen zwischen Dienstanbieter und Betreiber des Anmeldeservices festgelegt.

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

Bei Attributen mit der Anzahl 0/1 hängt es von der vertraglichen Absprache zwischen Dienstanbieter und Betreiber des Anmeldeservices 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 zu erwarten ist (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.

6.1 Angepasstes Datenmodell Person für Dienstanbieter

Attribut	Typ	Anz.	Freig. erford erlich	Bemerkung
referrer	String (UTF-8)	0..1	ja	ID der Gruppe im Quellsystem. Wird vom Quellsystem vergeben und muss im Quellsystem eindeutig sein.
name	<i>Strukturierung</i>	0/1	ja	
name.familiename	String (DIN 91379.A ¹)	0/1	ja	Familiename der Person
name.vorname	String (DIN 91379.A)	0/1	ja	Mehrere Vornamen werden durch Leerzeichen separiert.
name.initialenVorname	String (DIN 91379.A)	0/0..1	ja	Initial oder Initialen des Vornamens.
name.initialenFamiliename	String (DIN 91379.A)	0/0..1	ja	Initial oder Initialen des Familiennamens.
geburt	<i>Strukturierung</i>	0/0..1	ja	
geburt.datum	Datum (siehe Abschnitt 5.1)	0/0..1	ja	Geburtsdatum der Person
geburt.volljaehrig	String (Code)	0/0..1	ja	Gibt an, ob eine Person volljaehrig ist, ohne weitere Information ueber Geburtsdatum oder Alter zu

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

Attribut	Typ	Anz.	Freig. erford erlich	Bemerkung
				vermitteln. Referenz auf einen Code der Codeliste 12.7 Boolean Boolean.
geburt.geburtsort	String (DIN 91379.A)	0/0..1	ja	Geburtsort einer Person, bestehend aus Ortsnamen und Land, beispielsweise "Eindhoven, Niederlande". Wird kein Land angegeben, so wird "Deutschland" angenommen.
geschlecht	String (Code)	0/0..1	ja	Referenz auf einen Code der Codeliste 12.2 Geschlecht
lokalisierung	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.
vertrauensstufe	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 9: Datenmodell Person für Dienstleister

6.2 Angepasstes Datenmodell Personenkontext für Dienstleister

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 weiter gereicht.

Abhängig von Vereinbarungen mit dem Betrieb des Anmeldeservices ist es auch möglich, alle gespeicherten Personenkontexte mitzuliefern und die Auswahl erst in der Anwendung des Dienstes vorzunehmen. Dieses ist beispielsweise sinnvoll, wenn Informationen aus einem

Kontext in einen anderen Kontext übernommen werden sollen, beispielsweise bei Schulwechsel.

Attribut	Typ	Anz.	Attr.-Freig. erforderlich	Bemerkung
referrer	String (UTF-8)	0..1	ja	ID der Gruppe im Quellsystem. Wird vom Quellsystem vergeben und muss im Quellsystem eindeutig sein.
id	String	1	nein	ID des Personenkontexts
organisation	<i>Strukturierung</i>	0/1	ja	
organisation.id	String	0..1	nein	Die „Identifikation“ einer „Organisation“. Es wird vom Anmelde-Server sichergestellt, dass jede Organisation eine eindeutige ID besitzt.
organisation.kennung	String	0..1	nein	Die optionale Kennung (externe Identifikations-ID) einer „Organisation“ muss innerhalb eines Organisationstyps eindeutig sein. Der Wert ist eine Kennung der Organisation, die von einem externen Verantwortlichen vergeben und kontrolliert wird. Beispielhaft ist für Organisationen vom Typ „Schule“ die offizielle Schulnummer.
organisation.name	String (DIN 91379.B)	0..1	nein	Offizieller Name einer Organisation
organisation.typ	String (Code)	1	nein	Typ der Organisation Referenz auf einen Code der Codeliste 12.5 Organisationstyp
rolle	String (Code)	0/1	ja	Rolle der Person innerhalb der Organisation. Referenz auf einen Code der Codeliste 0

				Rolle
personenstatus	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
gruppen	Strukturierung	0/0..n	ja	Array aus Gruppen und dazugehörige Gruppenzugehörigkeiten
gruppen.gruppe.*	Komplexes Objekt	0/1	ja	Attribute einer einzelnen Gruppe entsprechend Datenmodell Gruppe 5.6.
gruppen.gruppenzugehörigkeit	Strukturierung	0/1	ja	
gruppen.gruppenzugehörigkeit.rollen	String (Code)	0/0..n	ja	Rollen der Person innerhalb der Gruppe. Liste von Rollen nach Codeliste 12.14 Gruppenrolle
loeschung	Strukturierung	0/0..1	nein	
loeschung.zeitpunkt	String (datetime)	1	nein	Datum und Uhrzeit an dem dieser Personenkontext gelöscht werden soll. Das Format des Löschezitpunktes ist yyyy-MM-dd'T'hh:mm'Z' als UTC-Zeitpunkt.

Datenmodell 10: Datenmodell Personenkontext für Dienstleister

6.3 Angepasstes Datenmodell Organisation für Dienstleister

Attribut	Typ	Anz.	Bemerkung
id	String (UTF-8)	1	ID der Organisation. Wird vom Anmeldeservice vergeben und ist eindeutig. Dieses Attribut ist unveränderbar (immutable).
kennung	String (UTF-8)	0..1	Die optionale Kennung (Identifikations-ID) einer "Organisation" muss innerhalb eines Organisationstyps eindeutig sein.

Attribut	Typ	Anz.	Bemerkung
name	String (DIN 91379.B)	0..1	Offizieller Name einer Organisation
typ	String (Code)	0..1	Typ der Organisation. Referenz auf einen Code der Codeliste 12.5 OrganisationstypOrganisationstyp

Datenmodell 11: Datenmodell Organisation für Dienstleister

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

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.

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 die Entwickler gedacht sind.

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 6: 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
400	09	„Datumsattribut hat einen ungültigen Wert“	„Datumsformat von Attribut X ist ungültig“, wobei X der Attributname ist.

Code	Sub-code	Titel	Beschreibung
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.

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, 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

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.“

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.“

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.“

500 (Interner 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 nachfolgenden Abschnitte definieren die Endpunkte der REST-API. Jeder Endpunkt stellt einen Zugriffspunkt auf eine Ressource dar. Es wird definiert, inwiefern die einzelnen CRUD-Operationen (CREATE, READ, UPDATE und DELETE) zugelassen sind und welche HTTP(S)-Methode dabei Verwendung findet. Insbesondere erfolgt jeweils die Definition der JSON-Objekte.

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/endpunkt`

Quellcode 7: URL-Struktur der Schnittstelle

Die API-Versionierung ermöglicht es den Entwicklern des Anmeldeservices 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://api.id-provider.schule/v1/organisation-info`

Quellcode 8: Beispiel der URL eines Schnittstellenendpunkts

Die konkrete URL für die Schnittstelle erhalten Sie von dem Betreiber des Anmeldeservice.

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 Anmeldeservice die client-Anwendung und nicht einen Benutzer. Hierfür verwenden client-Anwendungen den OAuth 2.0 Client Credentials Flow (OAuth 2.0 RFC 6749² "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 Anmeldeservices 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 Anmeldeservice.

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 6749³ „The OAuth 2.0 Authorization Framework“ muss der REST-API-Endpunkt für den „OAuth 2.0 Token Exchange“ (siehe IETF RFC 8693⁴ "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`).

² <https://datatracker.ietf.org/doc/html/rfc6749#section-4.4>

³ <https://datatracker.ietf.org/doc/html/rfc6749>

⁴ <https://datatracker.ietf.org/doc/html/rfc8693>

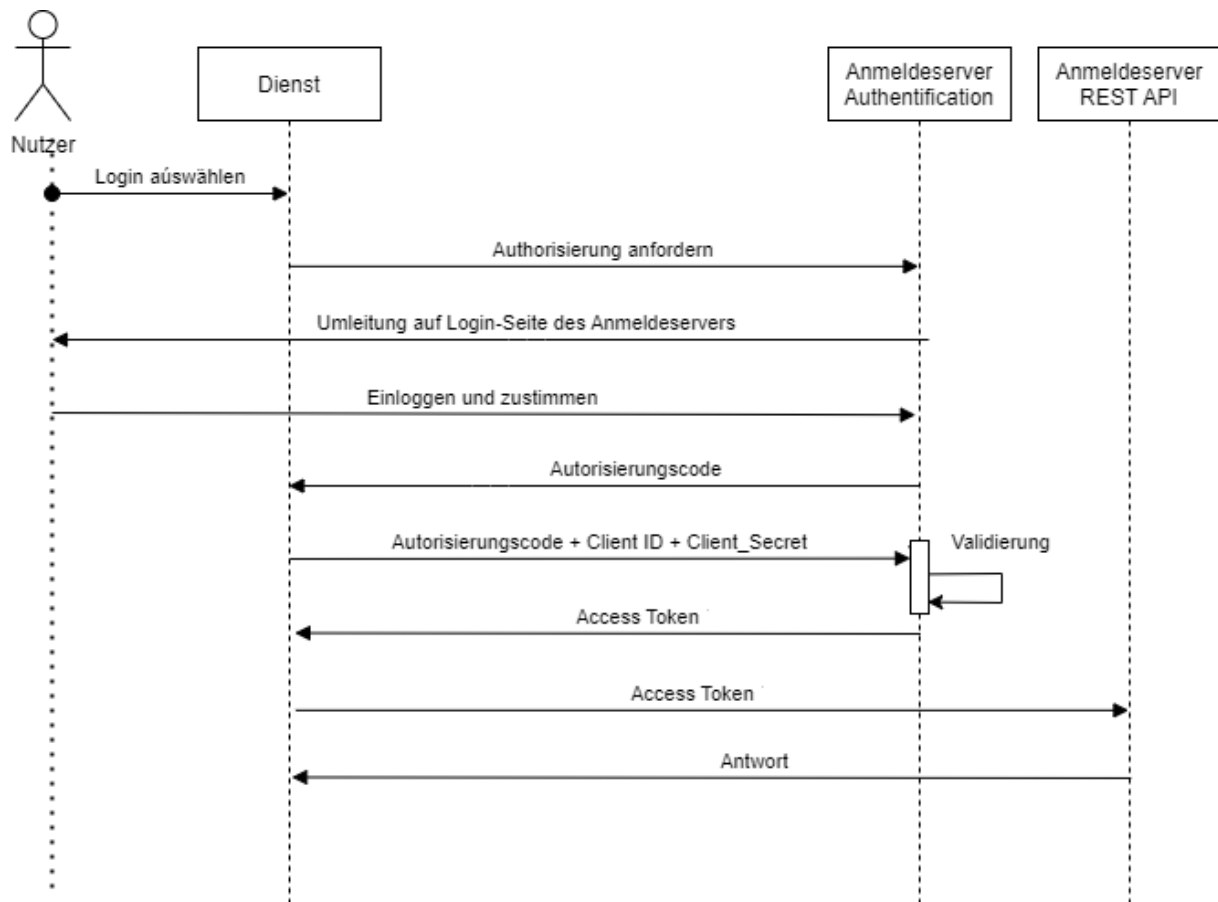


Abbildung 1: Sequenzdiagramm für die Autorisierung und anschließende Beispielabfrage von Personendaten im Sicherheitskontext einer angemeldeten nutzenden Person

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“⁵ 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 Grant⁶) 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

⁵ <https://datatracker.ietf.org/doc/html/rfc6750>

⁶ <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.

Für darauffolgende Anfragen der Client-Anwendungen gegen einen API-Endpunkt des SchulConneX-Servers 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 Basic⁷ 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“⁸ 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-Spezifikation⁹ entspricht.

Attribut Ebene 0	Typ	Anz.	Bemerkung
access_token	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 Frameworks ist spezifiziert in IETF RFC6749 Abschnitt 5.1 ¹⁰ .
expires_in	Integer	0..1	Die Gültigkeit des Tokens (<code>access_token</code>) in Sekunden ab Ausstellung. Das <code>access_token</code> beinhaltet ebenfalls die Dauer der Gültigkeit

⁷ <https://www.rfc-editor.org/rfc/rfc7617>

⁸ <https://datatracker.ietf.org/doc/html/rfc6749#section-4.4>

⁹ <https://datatracker.ietf.org/doc/html/rfc8693>

¹⁰ <https://datatracker.ietf.org/doc/html/rfc6749#section-5.1>

Attribut Ebene 0	Typ	Anz.	Bemerkung
			des Tokens. Dieser Wert ist <i>optional</i> , sofern ein access_token ausgestellt wurde. Beispielhaft steht der Wert 1800 für eine Gültigkeitsdauer von 30 Minuten.
refresh_token	String (byte)	0..1	Ein Token (refresh_token), welches vom Anmeldeservice auf die OAuth 2.0-Token-Exchange-Anfrage im Erfolgsfall <i>optional</i> ausgestellt wurde. Mit dem refresh_token kann beim Anmeldeservice ein neues access_token angefordert werden.
token_type	String (Enumerated)	0..1	Der token_type stellt die Information zur Verfügung, wie das access_token angewendet werden kann. Auswahl an token_type: <ul style="list-style-type: none"> • bearer¹¹ • mac¹² Im Kontext der Spezifikation OpenID Connect <i>muss</i> das Attribut token_type den Wert „Bearer“ führen.
id_token	String (byte)	0..1	Basierend auf der Spezifikation OpenID Connect beinhaltet die Token-Antwort einer OAuth 2.0-Token-Exchange-Anfrage zusätzlich ein Token (id_token ¹³), welches Informationen über die Authentifizierung einer nutzenden Person beinhaltet. Siehe 8.4.2
scope	String (byte)	0..1	Eine Liste von durch Leerzeichen getrennten Zeichenketten, die den Geltungsbereich ¹⁴ des Tokens spezifiziert.

Datenmodell 12: Access Token

¹¹ <https://datatracker.ietf.org/doc/html/rfc6750>

¹² <https://datatracker.ietf.org/doc/html/rfc6749#ref-OAuth-HTTP-MAC>

¹³ https://openid.net/specs/openid-connect-core-1_0.html#CodeIDToken

¹⁴ <https://datatracker.ietf.org/doc/html/rfc6749#section-3.3>

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)¹⁵. 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 1.0¹⁶ 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
iss	String (byte)	1	Der issuer identifier ist eine URL, welche denjenigen identifiziert, der die Inhalte herausgegeben hat.
sub	String (byte)	1	Der subject identifier ist eine Zeichenkette mit einer maximalen Länge von 255 ASCII-Zeichen und repräsentiert, wer lokal nutzt.
aud	String (byte)	1	Die audience (aud) claim listet die erwarteten Empfängerinnen und Empfänger des JWT auf. Im Falle eines ID Token enthält aud die Client-ID der Client-Anwendung, welche den ID Token angefordert hat.
exp	Integer	1	Das Attribut exp (expires) führt das Datum, zu wann das JWT Gültigkeit hat. Das Datum wird als Unix Epoch-Time geführt und ist ein Integer, welches die Sekunden seit dem 01. Januar 1970 angibt.
iat	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.
auth_time	Integer	0..1	Das optionale Attribut auth_time führt das Datum, zu welchem Zeitpunkt die nutzende Person authentifziert wurde. Das Datum wird als Unix Epoch-Time geführt und ist ein Integer, welches die Sekunden seit dem 01. Januar 1970 angibt.

¹⁵ <https://www.rfc-editor.org/rfc/rfc7519>

¹⁶ https://openid.net/specs/openid-connect-core-1_0.html

Attribut Ebene 0	Typ	Anz.	Bemerkung
nonce	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 Anmelde-Server diesen entsprechend zum Abgleich vorhält.
acr	String (byte)	0..1	Das optionale Attribut acr (Authentication Context Class Reference) referenziert die URI der Benutzerauthentifizierungsmethode ¹⁷ , welche erfüllt wurde.
amr	String (byte)	0..1	Das optionale Attribut amr (Authentication Method Reference) referenziert die Methode der Benutzerauthentifizierung ¹⁸ , welche verwendet wurde.
azp	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 13: ID Token

8.5 Vorbedingungen der Schnittstelle

Generell gilt für alle Anfragen an die API folgende Vorbedingung:

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 9: Beispiel eines Authorization Headers

¹⁷ ,https://openid.net/specs/openid-provider-authentication-policy-extension-1_0.html

¹⁸ <https://datatracker.ietf.org/doc/html/rfc8176>

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 10: 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://api.id-provider.schule/v1/person-info
```

Quellcode 11: 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.

Operation	Zugelassen?	HTTP-Methode
CREATE	Nein	PUT
READ	Ja	GET
UPDATE	Nein	PUT
DELETE	Nein	DELETE

Tabelle 7: Zugelassene Operationen auf dem Endpunkt „person-info“

READ

Es sind nur Anfragen mit der HTTP-GET-Methode für ein READ auf die Daten zugelassen. 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 Anmelde-Server 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
        "gruppenzugehoerigkeit": <Objekt>,
// Objekt vom Typ Gruppenzugehoerigkeit - einzig ausgegebenes Attribut
ist "rollen", ein Array von Gruppenrollen.
// Objekt vom Typ Gruppe

    ]
}

```

Quellcode 12: 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.

In späteren Versionen ist vorgesehen, 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 `id` des Personenkontexts, der gerade ausgewählt und somit für diese Anfrage aktiv ist. Diese Funktion ist jedoch aktuell nicht implementiert.

Die einzelnen Datenmodelle zu „Person“ und „Personenkontext“ sind in 5.3 bzw. 5.4 ausführlich beschrieben.

```

{
    "pid": "af3a88fc-d766-11ec-9d64-0242ac120002",
// id des ausgewählten Personenkontext
    "person": {
        "referrer": "a141e5ce-7a86-4f92-a600-923d83a5107f",
        "name": {
            "familienname": "Muster", // String

```



```

        "vorname": "Max" // String
    },
    "geburt": {
        "datum": "2010-01-01", // YYYY-MM-DD
        "volljaehrig": "ja"

    },
    "geschlecht": "d", // Laut Codeliste „d“ für „divers“
    "lokalisierung": "de-DE", // Laut Codeliste „de-DE“ für „Deutsch“
    "vertrauensstufe": "VOLL" // Laut Codeliste „VOLL“ für
„Vollständig“
    },
    "personenkontexte": [{
        "id": "af3a88fc-d766-11ec-9d64-0242ac120002",
        "referrer": "b232e5ce-7a86-4f92-a600-923d83a5208a",
        // id des Personenkontexts
        "organisation": {
            "id": "15685758-d18e-49c1-a644-f9996eb0bf08",
            // ID der Organisation
            "kennung": "NI_12345", // Kennung der Organisation
            "name": "Muster-Schule", // Name der Organisation
            "typ": "SCHULE" // Laut Codeliste für „Schule“
        },
        "rolle": "LERN", // Laut Codeliste „LERN“ für Lernende
        "personenstatus": "AKTIV", // Laut Codeliste für Personenstatus
        "gruppen": [
            {
                "gruppe": {
                    "id": <String>, // ID der Gruppe
                    "referrer": <String>, // Organisations spezifische 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 Gruppenbereich
        "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"
        "organisation": {
            "id": <string> ,// ID der Organisation
            "kennung": <string>, // Kennung der Organisation
            "name": <string> , // Name der Organisation
            "typ": <code> // Laut Codeliste für „Schule“
        },
    },
}
"gruppenzugehoerigkeit":
{
    "rollen": [<Code>, <Code>, <Code>...], // Rolle(n) in
der Gruppe
}
}
]
}]
}

```

Quellcode 13: 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 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 sicher zu stellen, dass im Anmeldeserve gelöschte Daten zeitnah auch bei Dienstanbietern zu löschen. Die Häufigkeit des Aufrufes ist mit dem Betreiber des Anmeldeserves zu vereinbaren.

9.2.1 Schnittstelle „personen-info“

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

`https://api.id-provider.schule/v1/personen-info`

Quellcode 14: Beispiel des Schnittstellen-Endpunkts für personen-info

Der Aufruf dieses Endpunktes erfolgt immer im Kontext des Diensts, basierend und authentisiert durch den Client Credentials Flow.

Operation	Zugelassen?	HTTP-Methode
CREATE	Nein	PUT
READ	Ja	GET
UPDATE	Nein	PUT
DELETE	Nein	DELETE

Tabelle 8: Zugelassene Operationen auf dem Endpunkt „personen-info“

READ

Es sind nur Anfragen mit der HTTP-GET-Methode für ein READ auf die Daten zugelassen. Als Antwort erfolgt eine Liste aus Datenobjekten, welche jeweils der Struktur einer Antwort der API `person-info` haben, jedoch nur drei Daten-Attribute enthält: `pid`, `id` und gegebenenfalls den Löschzeitpunkt.

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

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

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

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
  "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 15: 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 16: Beispiel einer JSON-Rückgabe zu personen-info

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.

https://subdomain.domain.org/subpaths/v1/personen

Quellcode 17: URL-Struktur des Endpunkts personen

Operation	Zugelassen?	HTTP-Methode
CREATE	Ja	POST
READ	Ja	GET
UPDATE	Nein	
DELETE	Nein	

Tabelle 9: 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 Anmeldeservice überprüft, ob der Wert des Filterparameters ohne Beachtung von Groß- und Kleinschreibung im Filter-Attribut beinhaltet ist (contains).
familienname	String	Möglichkeit, die Ergebnisliste der Anfrage nach dem Attribut "person.name.familienname" zu filtern. Der Anmeldeservice überprüft, ob der Wert des Filterparameters ohne Beachtung von Groß- und

Filter	Typ	Beschreibung
		Kleinschreibung im Filter-Attribut beinhaltet ist (contains).
vorname	String	Möglichkeit, die Ergebnisliste der Anfrage nach dem Attribut "person.name.vorname" zu filtern. Der Anmeldeservice überprüft, ob der Wert des Filterparameters ohne Beachtung von Groß- und Kleinschreibung im Filter-Attribut beinhaltet ist (contains).
sichtfreigabe	String	Erlaubt es Personen abhaengig davon aufzulisten, ob diese von einer anderen Organisation zum Lesen freigegeben wurden. 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.

Tabelle 10: Zugelassene Filter auf dem Endpunkt /v1/personen

CREATE

Mittels dieser Schnittstelle wird ein neuer Datensatz „Person“ erstellt.

Es sind nur Anfragen mit der HTTP-POST-Methode für ein CREATE 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 ausführlich beschrieben. Das Datenmodell „Person“ ist in 5.3 ausführlich 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 Anmeldeservice vergeben und ist eindeutig. Dieses Attribut ist unveränderbar (immutable).
mandant	nein	ID des Mandanten, dem die Personen zugeordnet ist. Wird vom Anmeldeservice vergeben und ist eindeutig. Dieser Wert ist eine Referenz auf einen Mandanten.
revision	nein	Revision der Personen. Wird vom Anmeldeservice mit der Erstellung des Datensatzes sowie Aktualisierung generiert. Dieser Wert kann nicht von Quellsystemen oder Diensten gesetzt werden.

Tabelle 11: 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 18: JSON-Struktur eines CREATE-Requests an /v1/personen

Beispiel Anfrage-Nutzdaten (Request Payload):

```

{
    "referrer" : "125",
    "name" :
    {
        "familienname" : "von Musterfrau",

```

```

        "vorname" : "Natalie",
        "initialenFamiliennamen" : "M",
        "initialenVorname" : "N",
        "sortierindex": "4"
    },
    "geburt" :
    {
        "datum" : "2005-05-01",
        "geburtsort" : "Berlin, Deutschland"
    },
    "geschlecht" : "w",
    "lokalisierung" : "de-DE",
    "vertrauensstufe" : "VOLL"
}

```

Quellcode 19: 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 20: 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 21: 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.

Es sind nur Anfragen mit der HTTP-GET-Methode für ein READ auf die Daten zugelassen.

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 ausführlich beschrieben. Das Datenmodell „Person“ ist in 5.3 ausführlich 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 Anmeldeservice 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 22: 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 23: 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 24: 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 25: URL-Struktur des Endpunkts personen/{personendatensatz.person.id}

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

Operation	Zugelassen?	HTTP-Methode
CREATE	Nein	
READ	Ja	GET
UPDATE	Ja	PUT
DELETE	Ja	DELETE

Tabelle 12: Zugelassene Operationen auf dem Endpunkt /v1/personen/{personendatensatz.person.id}

READ

Dieser Schnittstellenendpunkt liefert den Personendatensatz zur angeforderten ID.

Es sind nur Anfragen mit der HTTP-GET-Methode für ein READ 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 ausführlich beschrieben. Das Datenmodell „Person“ ist in 5.3 ausführlich 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 26: 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 27: 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.

Es sind nur Anfragen mit der HTTP-PUT-Methode für ein UPDATE 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 ausführlich beschrieben. Das Datenmodell „Person“ ist in 5.3 ausführlich beschrieben.

Es ist erforderlich, dass zur Aktualisierungsanfrage im bearbeiteten Personendatensatz das Attribut „personendatensatz.person.revision“ des zugrunde liegenden Personendatensatzes mitgeschickt wird. Der Anmeldeservice 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 Anmeldeservice 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 Anmeldeservice vergeben und ist eindeutig. Dieses Attribut ist unveränderbar (immutable).
mandant	Nein	ID des Mandanten, dem die Personen zugeordnet ist. Wird vom Anmeldeservice vergeben und ist eindeutig. Dieser Wert ist eine Referenz auf einen Mandanten.
revision	Ja (Wert des für die Änderung zugrunde liegendem „personendatensatz.person“)	Revision der Personen. Wird vom Anmeldeservice mit der Erstellung des Datensatzes sowie Aktualisierung generiert. Dieser Wert kann nicht von Quellsystemen oder Diensten gesetzt werden.

Tabelle 13: 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 28: 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 29: 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 30: 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 31: 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 32: 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.

Es sind nur Anfragen mit der HTTP-DELETE-Methode für ein 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 ausführlich beschrieben. Das Datenmodell „Person“ ist in 5.3 ausführlich beschrieben.

Es ist erforderlich, dass für eine Löschanfrage eines Personendatensatz das Attribut „personendatensatz.person.revision“ des zugrunde liegenden Personendatensatzes mitgeschickt wird. Der Anmeldeservice 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 33: JSON-Struktur eines DELETE-Requests an `/v1/personen/{personendatensatz.person.id}`

Beispiel Anfrage Nutzdaten (Request Payload):

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

Quellcode 34: 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 35: 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.

Operation	Zugelassen?	HTTP-Methode
CREATE	Ja	POST
READ	Ja	GET
UPDATE	Nein	
DELETE	Nein	

Tabelle 14: 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 Anmeldeservice 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 Anmeldeservice

Filter	Typ	Beschreibung
		<p>überprüft, ob der Wert des Filter-parameters mit dem Wert des Filter-Attribut 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. Der Anmeldeservice überprüft, ob der Wert des Filter-parameters mit dem Wert des Filter-Attribut übereinstimmt (equals).</p> <p>Dem Filterparameter liegt eine Codeliste Personenstatus zugrunde.</p>
sichtfreigabe	String	<p>Erlaubt es Personen abhaengig davon aufzulisten, ob diese von einer anderen Organisation zum Lesen freigegeben wurden. 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 15: 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}.

Es sind nur Anfragen mit der HTTP-POST-Methode für ein CREATE 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 ausführlich beschrieben.

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

Request

Beim Erstellen eines Personenkontext zu einem Personendatensatz wird immer die eigene Organisation per ID referenziert; die Organisation und dessen ID {organisation.id} wird vom Anmeldeservice aus dem access_token übernommen.

Die folgende Tabelle listet die Attribute eines Personkontext („personendatensatz.personkontexte“), welche von einem Quellsystem oder Dienst nicht verändert werden können.

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

revision	nein	Revision des Personenkontext. Wird vom Anmeldeservice mit der Erstellung des Datensatzes sowie Aktualisierung generiert. Dieser Wert kann nicht von Quellsystemen oder Diensten gesetzt werden.
----------	------	---

Tabelle 16: 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 36: 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 37: 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 38: JSON-Struktur einer CREATE-Response an /v1/personen/{personendatensatz.person.id}/personenkontexte

Die Organisation wird beim Erstellen eines Personenkontext vom Anmeldeservice 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 39: 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.

Es sind nur Anfragen mit der HTTP-GET-Methode für ein READ auf die Daten zugelassen. 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 ausführlich 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 40: JSON-Struktur einer READ-Response an /v1/personen/{personendatensatz.person.id}/personenkontexte

Die Organisation wurde beim Erstellen eines Personenkontexts vom Anmeldeservice 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 41: 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 42: 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 [/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 43: URL-Struktur des Endpunkts personenkontexte

Operation	Zugelassen?	HTTP-Methode
CREATE	Nein	
READ	Ja	GET
UPDATE	Nein	
DELETE	Nein	

Tabelle 17: Zugelassene Operationen 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:

Filter	Typ	Beschreibung
referrer	String	Möglichkeit, die Ergebnisliste der Anfrage nach dem Fremdschlüssel "referrer" zu filtern. Der Anmeldeservice 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 Anmeldeservice überprüft, ob der Wert des Filterparameters mit dem Wert des Filter-Attributs übereinstimmt (equals). Dem Filterparameter liegt eine Codeliste (Rolle) zugrunde.

Filter	Typ	Beschreibung
personenstatus	String (Code)	<p>Möglichkeit, die Ergebnisliste der Anfrage nach dem Attribut "personenkontext.personenstatus" zu filtern. Der Anmeldeservice 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 abhaengig davon aufzulisten, ob diese von einer anderen Organisation zum Lesen freigegeben wurden. 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 18: Zugelassene Filter auf dem Endpunkt /v1/personenkontexte

READ

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

Es sind nur Anfragen mit der HTTP-GET-Methode für ein READ auf die Daten 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 ausführlich beschrieben. Das Datenmodell „Person“ ist in 5.3 ausführlich beschrieben. Das Datenmodell „Personenkontext“ ist in 5.4 ausführlich 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 44: 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 – nur die 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"
    }
]

```

Quellcode 45: 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 46: 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.

10.1.5 [/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 47: URL-Struktur des Endpunkts personenkontexte/{personendatensatz.personenkontext.id}

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

Operation	Zugelassen?	HTTP-Methode
CREATE	Nein	
READ	Ja	GET
UPDATE	Ja	PUT
DELETE	Ja	DELETE

Tabelle 19: Zugelassene Operationen auf dem Endpunkt /v1/personenkontexte/{personendatensatz.personenkontext.id}

READ

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

Es sind nur Anfragen mit der HTTP-GET-Methode für ein READ auf die API `/v1/personenkontexte/{personendatensatz.personenkontext.id}` zum Abfragen eines Personenkontext eines 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 ausführlich beschrieben. Das Datenmodell „Person“ ist in 5.3 ausführlich beschrieben. Das Datenmodell „Personenkontext“ ist in 5.4 ausführlich 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 48: 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 49: 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 – nur die 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 50: 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}`.

Es sind nur Anfragen mit der HTTP-PUT-Methode für ein UPDATE 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 ausführlich beschrieben.

Es ist erforderlich, dass zur Aktualisierungsanfrage im bearbeiteten Personenkontext 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. 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 Anmeldeservice neu generiert.

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

Attribut	In den Anfrage Nutzdaten erforderlich?	Bemerkung
id	nein	ID des Personenkontext. Wird vom Anmeldeservice vergeben und ist eindeutig. Dieses Attribut ist unveränderbar (immutable).
mandant	nein	ID des Mandanten, dem die Personen zugeordnet ist. Wird vom Anmeldeservice vergeben und ist eindeutig. Dieser Wert ist eine Referenz auf einen Mandanten.
organisation	nein	
organisation.id	nein	ID der Organisation. Wird vom Anmeldeservice 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 führen ein Update auf diesen Personenkontext zu einer Fehlermeldung, da ein Sichtfreigabe nur das Lesen des Personenkontexts erlaubt.
revision	Ja (Wert des für die Änderung zugrunde liegendem Personenkontext	Revision des Personenkontexts. Wird vom Anmeldeservice 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/{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 Personenkontext 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 51: JSON-Struktur eines UPDATE-Requests an /v1/personenkontexte/{id}

Beispiel Anfrage-Nutzdaten (Request Payload):

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

Quellcode 52: 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 53: 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 54: 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}`.

Es sind nur Anfragen mit der HTTP-DELETE-Methode für ein 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 ausführlich 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 Personenkontext 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 ein Personenkontext bereits von anderen Systemen (Quellsysteme oder Dienstanbieter) genutzt wurde. In dem Fall muss die Löschung auf einem anderen Weg erfolgen, um sicherzustellen, dass andere Dienste über die Löschung informiert werden und eventuell existierende lokale Instanzen der Information löschen können.

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 57: URL-Struktur des Endpunkts organisationen

Operation	Zugelassen?	HTTP-Methode
CREATE	Nein	
READ	Ja	GET
UPDATE	Nein	
DELETE	Nein	

Tabelle 21: Zugelassene Operationen 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:

Filter	Typ	Beschreibung
kennung	String	Möglichkeit, die Ergebnisliste der Anfrage nach "kennung" zu filtern. Der Anmeldeservice 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 Anmeldeservice 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. Der Anmeldeservice 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 22: Zugelassene Filter auf dem Endpunkt /v1/organisationen

READ

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

Es sind nur Anfragen mit der HTTP-GET-Methode für ein READ auf die Daten zugelassen. 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 ausführlich beschrieben.

Request

Die Anfrage erfordert keine Nutzdaten (Request Payload).

Response

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

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

Quellcode 58: 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 59: 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 60: 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 61: URL-Struktur des Endpunkts organisationen/{organisation.id}

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

Operation	Zugelassen?	HTTP-Methode
CREATE	Nein	
READ	Ja	GET
UPDATE	Nein	
DELETE	Nein	

Tabelle 23: Zugelassene Operationen auf dem Endpunkt /v1/organisationen/ {organisation.id}

READ

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

Es sind nur Anfragen mit der HTTP-GET-Methode für ein READ 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 ausführlich 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
  "typ": <Code>, // Siehe Code-Liste "Organisation / Typ"
  "traegerschaft": <Code>, // Siehe Code-Liste "Trägerschaft"
} // Siehe "Organisation"
```

Quellcode 62: 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",
  "typ": "SCHULE",
  "traegerschaft": "02"
}
```

Quellcode 63: 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 [/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 64: URL-Struktur des Endpunkts organisation-info

Operation	Zugelassen?	HTTP-Methode
CREATE	Nein	
READ	Ja	GET
UPDATE	Nein	
DELETE	Nein	

Tabelle 24: Zugelassene Operationen auf dem Endpunkt /v1/organisation-info

READ

Dieser Schnittstellenendpunkt gibt die Informationen der anfordernden Organisation zurück.

Es sind nur Anfragen mit der HTTP-GET-Methode für ein READ 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 ausführlich 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
  "typ": <Code>, // Siehe Code-Liste "Organisation / Typ"
  "traegerschaft": <Code>, // Siehe Code-Liste "Trägerschaft"
} // Siehe "Organisation"
```

Quellcode 65: 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",
  "typ": "SCHULE",
  "traegerschaft": "02"
}
```

Quellcode 66: 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 67: URL-Struktur des Endpunkts Gruppen

Operation	Zugelassen?	HTTP-Methode
CREATE	Ja	POST
READ	Ja	GET
UPDATE	Nein	
DELETE	Nein	

Tabelle 25: Zugelassene Operationen auf dem Endpunkt /v1/gruppen

CREATE

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

Es sind nur Anfragen mit der HTTP-POST-Methode für ein CREATE 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 ausführlich beschrieben. Die Gruppen-ID, die Mandanten-ID, die Organisations-ID und die Revisionsnummer werden vom SchulConneX-Server 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 12.6 Jahrgangsstufe
  "faecher": [<faecher>, <faecher>, <faecher>...]
  // Objekte mit Faecher Attributen, siehe Beschreibung "Gruppe"
  "referenzgruppen": [{
    "id": <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 68: 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 69: 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": [{  
    "id": "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 70: 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.

Es sind nur Anfragen mit der HTTP-GET-Methode für ein READ auf die Daten zugelassen. Die Antwort-Nutzdaten (Response Payload) beinhalten ein Array von JSON-Objekten vom Datentyp Gruppendatensatz, sofern Einträge existieren. Das Datenmodell „Gruppendatensatz“ ist in 5.8 ausführlich 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 Anmeldeservice, ob der Wert des Filterparameters ohne Beachtung von Groß- und Kleinschreibung im Filter-Attribut beinhaltet ist (contains). Bei Filtern vom Typ „Code“ überprüft der Anmeldeservice, 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 sowohl Niederländisch, Italienisch und Spanisch gelehrt wird.

Ein Sonderfall ist hier bei dem Filter „faecher“ gegeben. Ein Suchparameter bei diesem Filter matched sowohl bei dem Unterattribut „kennung“ als auch dem Unterattribut „text“. Wird in einer Gruppe beispielsweise das Standard-Fach English unterrichtet und auch das Thema „Filmschnitt“, das nicht in der Fäche-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	Erlaubt es Gruppen abhaengig davon aufzulisten, ob diese von einer anderen Organisation zum Lesen freigegeben wurden. 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.

Tabelle 26: 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 71: 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 12.13 Gruppentyp
      "bereich": <Code>, // Siehe Code-Liste 12.10 Gruppen
      "optionen": [<Code>, <Code>, <Code>...], // Siehe Code-Liste 12.12
      Gruppenoption
      "differenzierung": <Code>, // Siehe Code-Liste 12.11
      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": [{
        "id": <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 72: 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 73: 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.

<code>https://subdomain.domain.org/subpaths/v1/gruppen/{gruppe.id}</code>

Quellcode 74: URL-Struktur des Endpunkts `gruppen/{gruppe.id}`

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

Operation	Zugelassen?	HTTP-Methode
CREATE	Nein	
READ	Ja	GET
UPDATE	Ja	PUT
DELETE	Ja	DELETE

Tabelle 27: Zugelassene Operationen auf dem Endpunkt `/v1/gruppen/{gruppe.id}`

READ

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

Es sind nur Anfragen mit der HTTP-GET-Methode für ein READ 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 ausführlich 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 75: 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": [{
        "id": "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"
},
"gruppenzugehoerigkeiten": [{
    "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 76: 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}`.

Es sind nur Anfragen mit der HTTP-PUT-Methode für ein UPDATE 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 ausführlich beschrieben.

Es ist erforderlich, dass zur Aktualisierungsanfrage in der bearbeiteten Gruppe das Attribut „gruppe.revision“ der zugrunde liegenden Gruppe mitgeschickt wird. Der Anmeldeservice 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 Anmeldeservice neu generiert.

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

Attribut	Bemerkung
id	ID der Gruppe. Wird vom Anmeldeservice vergeben und ist eindeutig. Dieses Attribut ist unveränderbar (immutable).
mandant	Mandanten-ID der Gruppe. Wird vom Anmeldeservice 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 führen, da ein Sichtfreigabe nur das Lesen der Gruppe erlaubt.
revision	Revision der Gruppe. Wird vom Anmeldeservice mit der Erstellung des Datensatzes sowie Aktualisierung generiert. Dieser Wert kann nicht von Quellsystemen oder Diensten gesetzt werden.

Tabelle 28: 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": [{
    "id": <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 77: 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": [{
    "id": <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 78: 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": [{
    "id": "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": "2"
}

```

```
}
```

Quellcode 79: 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}`.

Es sind nur Anfragen mit der HTTP-DELETE-Methode für ein 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 ausführlich beschrieben.

Es ist erforderlich, dass für eine Löschanfrage einer Gruppe das Attribut „`gruppe.revision`“ der zugrunde liegenden Gruppe mitgeschickt wird. Der Anmeldeservice 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 80: JSON-Struktur eines DELETE-Requests an `/v1/gruppen/{gruppe.id}`

Beispiel Anfrage-Nutzdaten (Request Payload):

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

Quellcode 81: 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 82: URL-Struktur des Endpunkts gruppen/{gruppe.id}/gruppenzugehoerigkeiten

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

Operation	Zugelassen?	HTTP-Methode
CREATE	Ja	POST
READ	Ja	GET
UPDATE	Nein	
DELETE	Nein	

Tabelle 29: Zugelassene Operationen auf dem Endpunkt /v1/gruppen/{gruppe.id}/gruppenzugehoerigkeiten

CREATE

Dieser Schnittstellenendpunkt erstellt eine Gruppenzugehörigkeit zur angegebenen Gruppe per ID {gruppe.id}.

Es sind nur Anfragen mit der HTTP-POST-Methode für ein CREATE 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 ausführlich beschrieben.

Request

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

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

Tabelle 30: 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 83: 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 84: 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 85: 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 86: 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.

Es sind nur Anfragen mit der HTTP-GET-Methode für ein READ 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 ausführlich 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 Anmeldeservice, ob der Wert des Filterparameters ohne Beachtung von Groß- und Kleinschreibung im Filter-Attribut beinhaltet ist (contains). Bei Filtern vom Typ „Code“ überprüft der Anmeldeservice, 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.

Tabelle 31: 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 87: 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 88: 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 89: URL-Struktur des Endpunkts gruppenzugehoerigkeiten

Operation	Zugelassen?	HTTP-Methode
CREATE	Nein	
READ	Ja	GET
UPDATE	Nein	
DELETE	Nein	

Tabelle 32: Zugelassene Operationen auf dem Endpunkt `/v1/gruppenzugehoerigkeiten`

READ

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

Es sind nur Anfragen mit der HTTP-GET-Methode für ein READ auf die Daten zugelassen. 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 umfasst standardmäßig für das Attribut {gruppe } mit dem Datentyp Gruppe nur die ID 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 Anmeldeservice, ob der Wert des Filterparameters ohne Beachtung von Groß- und Kleinschreibung im Filter-Attribut beinhaltet ist (contains). Bei Filtern vom Typ “Code” überprüft der Anmeldeservice, 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 33: 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 90: 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 91: 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 92: URL-Struktur des Endpunkts gruppenzugehoerigkeiten/{gruppenzugehoerigkeit.id}

Der Pfad-Parameter {gruppenzugehoerigkeit.id} bezieht sich auf die ID der Gruppenzugehörigkeit.

Operation	Zugelassen?	HTTP-Methode
CREATE	Nein	
READ	Ja	GET
UPDATE	Ja	PUT
DELETE	Ja	DELETE

Tabelle 34: Zugelassene Operationen auf dem Endpunkt /v1/gruppenzugehoerigkeiten/{gruppenzugehoerigkeit.id}

READ

Dieser Schnittstellenendpunkt gibt die Gruppenzugehörigkeit zur angegebenen {gruppenzugehoerigkeit.id} zurück.

Es sind nur Anfragen mit der HTTP-GET-Methode für ein READ 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 ausführlich 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
    Gruppensoption
    "differenzierung": <Code>, // Siehe Code-Liste 12.11
    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 93: 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 94: 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}`.

Es sind nur Anfragen mit der HTTP-PUT-Methode für ein UPDATE 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 ausführlich beschrieben.

Es ist erforderlich, dass zur Aktualisierungsanfrage in der bearbeiteten Gruppenzugehörigkeit das Attribut `“revision”` der zugrunde liegenden Gruppenzugehörigkeit mitgeschickt wird. Der Anmeldeservice 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 Anmeldeservice neu generiert.

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

Attribut	Bemerkung
id	ID der Gruppenzugehörigkeit. Wird vom Anmeldeservice vergeben und ist eindeutig. Dieses Attribut ist unveränderbar (immutable).
revision	Revision der Gruppenzugehörigkeit. Wird vom Anmeldeservice mit der Erstellung des Datensatzes sowie Aktualisierung generiert. Dieser Wert kann nicht von Quellsystemen oder Diensten gesetzt werden.

Tabelle 35: 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 95: 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 96: 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 97: 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}`.

Es sind nur Anfragen mit der HTTP-DELETE-Methode für ein 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 ausführlich beschrieben.

Es ist erforderlich, dass für eine Löschanfrage einer Gruppenzugehörigkeit das Attribut „revision“ der zugrunde liegenden Gruppenzugehörigkeit mitgeschickt wird. Der Anmeldeservice 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 98: JSON-Struktur eines DELETE-Requests an `/v1/gruppenzugehoerigkeiten/{gruppenzugehoerigkeit.id}`

Beispiel Anfrage-Nutzdaten (Request Payload):

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

Quellcode 99: 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 100: URL-Struktur des Endpunkts codelisten

Operation	Zugelassen?	HTTP-Methode
CREATE	Nein	
READ	Ja	GET
UPDATE	Nein	
DELETE	Nein	

Tabelle 36: Zugelassene Operationen 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 101: 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 102: URL-Struktur des Endpunkts codelisten/{codeliste}

Operation	Zugelassen?	HTTP-Methode
CREATE	Nein	
READ	Ja	GET
UPDATE	Nein	
DELETE	Nein	

Tabelle 37: Zugelassene Operationen 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 103: 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 104: URL-Struktur des Endpunkts versionen

Operation	Zugelassen?	HTTP-Methode
CREATE	Nein	
READ	Ja	GET
UPDATE	Nein	
DELETE	Nein	

Tabelle 38: Zugelassene Operationen auf dem Endpunkt /v1versionen

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 Nummern. 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 105: 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 koennen diese zur aktuellen Version mitgeliefert werden. Dabei wird immer der URL Pfad bis einschliesslich der Versionsnummer und einem abschliessenden Schraegstrich 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-sensitive). Die Organisationsarten “SCHULE”, “schule”, “Schule” oder auch “SchuLe” sind als identisch anzunehmen.

12.1 Personenstatus

Code	Bezeichnung
AKTIV	Aktiv

Codeliste 2: Personenstatus

12.2 Geschlecht

Code	Bezeichnung
m	männlich
w	weiblich
d	divers
x	Keine Angabe

Codeliste 1: Geschlecht

12.3 Rolle

Code	Bezeichnung
LERN	Lernende/r
LEHR	Lehrende/r
EXTERN	Externe Person
ORGADMIN	Organisationsadministrator
LEIT	Organisationsleitung
SYSADMIN	Systemadministrator

Codeliste 3: Rolle

12.4 Vertrauensstufe

Code	Bezeichnung	Bemerkung
KEIN	Keine	Der Schlüsselbesitzer ist nicht vertrauenswürdig. Diese Vertrauensstufe wird meist für Test- oder Beispiel-Nutzer vergeben.
UNBE	Unbekannt	Der Schlüsselbesitzer ist unbekannt. Diese Vertrauensstufe wird beispielsweise für selbstregistrierte Nutzer vergeben.
TEIL	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.
VOLL	Verifiziert	Der Schlüsselbesitzer ist eindeutig und formal identifiziert (beispielsweise durch Vorlage des Personalausweises).

Codeliste 4: Vertrauensstufe

12.5 Organisationstyp

Code	Bezeichnung
SCHULE	Schule
ANBIETER	Anbieter
SONSTIGE	Sonstige Organisation / Einrichtungen

Codeliste 5: Organisationstyp

12.6 Jahrgangsstufe

Code	Bezeichnung
01	Jahrgangsstufe 1
02	Jahrgangsstufe 2
03	Jahrgangsstufe 3
04	Jahrgangsstufe 4
05	Jahrgangsstufe 5
06	Jahrgangsstufe 6
07	Jahrgangsstufe 7
08	Jahrgangsstufe 8
09	Jahrgangsstufe 9
10	Jahrgangsstufe 10

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 5646¹. 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
Pflicht	Pflichtunterricht
Wahl	Wahlunterricht
Wahlpflicht	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
G	G-Kurs
E	E-Kurs
Z	Z-Kurs
gA	grundlegendes Anforderungsniveau
eA	erhöhtes Anforderungsniveau

Codeliste 11: Gruppendifferenzierung

12.12 Gruppenoption

Code	Bezeichnung
01	bilingual
02	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

2024-2	2. Halbj. 24/25	2025-02-01	2025-07-31	HJ
2025	Schuljahr 2025/26	2025-08-01	2026-07-31	SJ
2025-1	1. Halbj. 25/26	2025-08-01	2026-01-31	HJ
2025-2	2. Halbj. 25/26	2026-02-01	2026-07-31	HJ
2026	Schuljahr 2026/27	2026-08-01	2027-07-31	SJ
2026-1	1. Halbj. 26/27	2026-08-01	2027-01-31	HJ
2026-2	2. Halbj. 26/27	2027-02-01	2027-07-31	HJ
2027	Schuljahr 2027/28	2027-08-01	2028-07-31	SJ
2027-1	1. Halbj. 27/28	2027-08-01	2028-01-31	HJ
2027-2	2. Halbj. 27/28	2028-02-01	2028-07-31	HJ

Codeliste 15: Lernperiode

12.16 Lernperiodentyp

Code	Bezeichnung
SJ	Schuljahr
HJ	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
BI	Biologie
CH	Chemie
CI	Chinesisch
DE	Deutsch
DS	Darstellendes Spiel
EK	Erdkunde

EN	Englisch
FR	Französisch
GR	Griechisch
NL	Niederländisch
IT	Italienisch
SN	Spanisch
KU	Kunst
LA	Latein
RS	Russisch
GE	Geschichte
PO	Politik
PW	Politik/Wirtschaft
RE	Evangelische Religion
RI	Islamische Religion
RK	Katholische Religion
SP	Sport
SU	Sachunterricht
TE	Technik
TG	Textiles Gestalten
WE	Gestaltendes Werken
WN	Werte und Normen
WS	Wirtschaft
DA	Deutsch als Zweitsprache

MA	Mathematik
HW	Hauswirtschaft
MU	Musik
PA	Pädagogik
PH	Physik
IF	Informatik
AW	Arbeit-Wirtschaft-Technik
GL	Gesellschaftslehre
PWI	Profil Wirtschaft
PTE	Profil Technik
PGUS	Profil Gesundheit und Soziales
NAT	Naturwissenschaften

Codeliste 17: Fächerkanon

12.18 Bildungsgang

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
GS	Grundschule
HS	Hauptschule
RS	Realschule
GY-SEK-I	Gymnasium Sekundarstufe I

Codeliste 18: Bildungsgang

12.19 Bildungsziel

Hinweis: In Niedersachsen ist diese Codeliste identisch mit der Codeliste Bildungsgang. Bei Verwendung dieser API in anderen Bundesländern der können die Codelisten aber auch voneinander abweichen.

Code	Bezeichnung
GS	Grundschule
HS	Hauptschule
RS	Realschule
GY-SEK-I	Gymnasium Sekundarstufe I

Codeliste 18: Bildungsziel

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 Dienstanbieters, 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 Anmeldeservice dar.
Natürliche Person	Siehe: Person
OAuth	Bei OAuth (Open Authorization) handelt es sich um ein offenes Sicherheitsprotokoll für die Token-basierte Autorisierung und Authentifizierung

Begriff	Erklärung
	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.
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.
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-Server 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.
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
id	ID of the organisation. Provided by the server and unique.
kennung	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 organization type
name	Official name of the organisation
namensergaenzung	Addition to the organisation name (such as a previous name or a suffix that isn't part of the name)
kuerzel	Short name of the organisation.
typ	Organisation type. See codelist 12.5 Organisationstyp
traegerschaft	Owner type of the organisation. See Codelist 12.8 Trägerschaft

14.1.3 Person (eng.)

Attribute	Description
id	ID of the person. Provided by the server and unique.
referrer	"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.
mandant	ID of the organisation the person is attached to. This is mainly a legal relationship, in the sense that this is the organisation responsible for maintaining the data for that person. Persons can be attached contextually to other organisations with the 'Personenkontext'.
stammorganisation	A person can belong administratively to one organisation, but might be delegated to another organisation. The 'stammorganisation' is the ID of the organisation the person primarily belongs to.
name	Name entity
name.familiennname	Surname of the person
name.vorname	Given name(s) of the person
name.initialenFamiliennname	Initial or initials of the surname
name.initialenVorname	Initial or initials of the given names
name.rufname	Name the person commonly called by.
name.titel	Title of the person
name.anrede	List of forms of address (e.g. Mr.)
name.namenspräfix	List of name prefixes (most often formal titles)
name.namenssuffix	List of name suffixes

Attribute	Description
name.sortierindex	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.
geburt	Birth information entity
geburt.datum	Date of birth
geburt.geburtsort	Place of birth, usually town and country. If no country is provided, Germany is assumed.
geschlecht	Sex of person. Refer to codelist 12.2 Geschlecht
lokalisierung	Lokalisation according to RFC5646. If none is provided „de-de“ is assumed. Refer to codelist 12.9 Lokalisierung.
vertrauensstufe	Describes the level of trust in the data provided. Refer to codelist 12.4 Vertrauensstufe.
auskunftssperre	Blocked information. Is set to "JA" (yes) if access to information of this person is blocked. Refer to codelist 12.7 Boolean.
revision	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.

14.1.4 Personenkontext (eng.)

The person context describes the role of a person within an organisation.

A person can have multiple person context. 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
id	ID of the person context. Provided by the server and unique
referrer	"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.
mandant	ID of the organisation the person context is attached to.
organisation	Organisation the person has a role in. See 14.1.2 Organisation
rolle	Role the person has in the organisation. Refer to codelist 12.3 Rolle.
personenstatus	Status the person has in relation to a role. See codelist 12.1 Personenstatus.
jahrgangsstufe	School grade or form a person is currently in or associated with. Refer to codelist 12.6 Jahrgangsstufe.
sichtfreigabe	Boolean value that denotes whether the data is only visible due to an other organisation providing read access.
loeschung.zeitpunkt	Date and time when the personkontext is to be deleted.
revision	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
person	Person. (See 5.3 Personen)
personenkontexte	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
id	ID of the group. Provided by the server and unique
mandant	ID of the organisation that owns the group dataset
orgid	ID of the organisation that the group is attached to. Often the same as "mandant", but not always so.
referrer	"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.
bezeichnung	Describes the group
thema	Keywords describing the theme of the group, short text, ideally less than 35 characters
beschreibung	Additional optional description of the purpose of the group.
typ	Type of the group. Refer to codelist 12.13 Gruppentyp.

bereich	Additional group qualifier (mandatory, optional, basic, advanced...) Refer to codelist 12.10 Gruppenbereich.
optionen	Additional group options. Usually referring to language used (bilingual, native language) Refer to codelist 12.12 Gruppenoption.
differenzierung	Specific differentiation in some educational branches. Refers to codelist 12.11 Gruppendifferenzierung.
bildungsziele	Educational branch. Refers to codelist 12.1812.19 Bildungsgang.
jahrgangsstufen	Form or course level. Refers to codelist 12.6 Jahrgangsstufe.
faecher	Array to list the topics covered in the group.
faecher.code	Topic covered. Refers to codelist 12.17 Fächerkanon .
referenzgruppen	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.
referenzgruppen.id	ID of the reference group
referenzgruppe.rollen	Roles in the reference group ("Gruppenrollen") to be selected for the group. 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.)
laufzeit	Defines the start and end date of a group. 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
laufzeit.von	Start date of the group as a date.

laufzeit.vonlernperiode	Start date of the group as a reference to a teaching period. Refer to codelist 12.15 Lernperiode
laufzeit.bis	End date of the group as a date.
laufzeit.bislernperiode	End date of the group as a reference to a teaching period. Refer to codelist 12.15 Lernperiode
sichtfreigabe	Boolean value that denotes whether the data is only visible due to an other organisation providing read access.
Revision	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.

14.1.7 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
id	ID of the group membership. Provided by the server and unique
referrer	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.
ktid	ID of the person context associated with the group.
rollen	Role of a person within the group. Refer to codelist 12.14 Gruppenrolle
von	Begin of the group membership
bis	End of the group membership
revision	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.
--	---

14.1.8 Gruppendatensatz (eng.)

The "group data set" is a wrapper object to link a group to a number of group memberships.

Attribute	Description
gruppe	Group. (See 5.6 Gruppe)
gruppenzugehoerigkeiten	Array of group memberships (See 5.7 Gruppenzugehörigkeit)

14.1.9 Freigabe (eng.)

A "Freigabe" enables organisations to make information available as "read only" information to other organisations..

Attribute	Description
id	ID of the data clearance
orgid	ID der organisation for which read access to the data has been cleared.
objid	ID of the element (person context or group) for which read access is been provided
von	Begin of the access period
bis	End of the access period
revision	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.

14.1.10 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
code	Code of the teaching period
bezeichnung	Description of teaching period, for example „Schuljahr 2023/2024“
typ	Type of the teaching period. Refers to codelist 12.16 Lernperiodentyp
beginn	Beginning date of teaching period
ende	Ending date of teaching period

14.2 Codelists (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 codelists 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 codelists, a short description of all lists is provided, though a translation of the individual codes is not given.

Codelist name	Description
Personenstatus	This is used to describe whether the dataset for a person is currently active
Geschlecht	Options for the sex of a person (male/female/divers/not-given)

Rolle	The role within an organisation, such as pupil, teacher or administrator)
Vertrauensstufe	Verification level of the owner of a (digital access) key, from untrusted or unknown to fully trusted.
Organisationstyp	Type of an organisation. Usually a school, but can be an external provider of educational content.
Jahrgangsstufe	Describes which form or level (currently year 1 to 10) a pupil is in at course is suitable for.
Boolean	Defines what terms are used for boolean values ("ja" for true and "nein" for false)
Trägerschaft	The 'ownership' of an organisation. Describes whether it is owned by the state, the county, the church or privately owned.
Lokalisierung	Provides localisation information. Usually "de-de" to denote German language content.
Gruppenbereich	Denotes whether participation in a group is optional ("Wahl"), mandatory ("Pflicht") or a combination of both for its members.
Gruppendifferenzierung	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.
Gruppenoption	Currently there are only two options available, one being "bilingual" and the other "taught in (pupil's) naive language".
Gruppentyp	Classification of a group as "class", "course" or "other". Classes are generally mandatory, while courses are often optional.
Gruppenrolle	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.
Lernperiode	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 14.1.10 Lernperiode.
Lernperiodentyp	Type (essentially the length) of a teaching period. Year or half-year.
Fächerkanon	Subjects that can be taught in a class or course. Such as biology, french or natural sciences.
Bildungsgang	Type of school, based on likely qualification level upon leaving.

15 Anlage: Dokumenthistorie

Datum	Autor*in	Organisation	Version	Status	Änderungen
22.03.2022	Christopher Krauss	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	FOKUS	0.000.002 - 0.000.004	Interner Arbeitsentwurf	Ausarbeitung einer minimalen Spezifikation für Service Provider
06.05.2022	Christopher Krauss	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	FOKUS	0.001.002	Kommentierbarer Arbeitsentwurf	Einarbeiten und Auflösen der Kommentare. Treffen finaler Festlegungen.
20.05.2022	Christopher Krauss	FOKUS	0.001.003	Veröffentlichungskandidat	Bereitstellung des Veröffentlichungskandidaten der Spezifikation für Service Provider
04.07.2022 – 08.07.2022	Christopher Krauss	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 Codetabellen 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 /gruppenzugehoerigkeiten. 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 Löschzeitpunkt
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 „SchulConnectX“, Ersetzen „Anmeldeserver“ durch „SchulConneX Server“, 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