

# Inhaltsverzeichnis

0.1	Datenmodell-Beispiele . . . . .	1
0.1.1	Benutzer (Datenmodelle) . . . . .	1
<b>1</b>	<b>Einleitung und Vorüberlegungen</b>	<b>11</b>
1.1	Benutzerrollen . . . . .	11
<b>2</b>	<b>Autorisierung und Authentifizierung</b>	<b>13</b>
2.1	OAuth2 . . . . .	13
2.2	OpenID Connect . . . . .	14
2.3	Einschränkung der Daten . . . . .	14
<b>3</b>	<b>REST-API-Definition</b>	<b>16</b>
3.1	Schnittstellen für Schulfächer . . . . .	16
3.1.1	Endpunkt in der REST-API: /api/school-subjects . . . . .	16
3.1.1.1	READ . . . . .	16
3.2	Schnittstellen für Schuljahre . . . . .	17
3.2.1	Endpunkt in der REST-API: /api/school-years . . . . .	17
3.2.1.1	READ . . . . .	17
3.3	Schnittstellen für Schulen . . . . .	18
3.3.1	Endpunkt in der REST-API: /api/schools . . . . .	18
3.3.1.1	READ . . . . .	19
3.3.2	Endpunkt in der REST-API: /api/schools/\$id . . . . .	19
3.3.2.1	READ . . . . .	19
3.3.3	Endpunkt in der REST-API: /api/schools/\$id/users . . . . .	20
3.3.3.1	READ . . . . .	20
3.3.4	Endpunkt in der REST-API: /api/schools/\$id/classes . . . . .	21
3.3.4.1	READ . . . . .	21
3.3.5	Endpunkt in der REST-API: /api/schools/\$id/subjects . . . . .	22
3.3.5.1	READ . . . . .	22
3.4	Schnittstellen für Benutzer . . . . .	23
3.4.1	Endpunkt in der REST-API: /api/users . . . . .	23
3.4.1.1	READ . . . . .	23
3.4.2	Endpunkt in der REST-API: /api/users/\$id . . . . .	24
3.4.2.1	READ . . . . .	24
3.4.3	Endpunkt in der REST-API: /api/users/\$id/memberships . . . . .	25
3.4.3.1	READ . . . . .	25

3.4.4	Endpunkt in der REST-API: /api/users/\$id/classes . . . . .	26
3.4.4.1	READ . . . . .	27
3.4.5	Endpunkt in der REST-API: /api/users/\$id/subjects . . . . .	27
3.4.5.1	READ . . . . .	28
3.4.6	Endpunkt in der REST-API: /api/users/\$id/children . . . . .	28
3.4.6.1	READ . . . . .	28
3.4.7	Endpunkt in der REST-API: /api/users/\$id/guardians . . . . .	29
3.4.7.1	READ . . . . .	29
3.5	Schnittstellen für Unterrichtsfächer an Schulen . . . . .	30
3.5.1	Endpunkt in der REST-API: /api/subjects . . . . .	30
3.5.1.1	READ . . . . .	30
3.5.2	Endpunkt in der REST-API: /api/subjects/\$id . . . . .	31
3.5.2.1	READ . . . . .	31
3.5.3	Endpunkt in der REST-API: /api/subjects/\$id/classes . . . . .	32
3.5.3.1	READ . . . . .	32
3.5.4	Endpunkt in der REST-API: /api/subjects/\$id/students . . . . .	33
3.5.4.1	READ . . . . .	33
3.5.5	Endpunkt in der REST-API: /api/subjects/\$id/teachers . . . . .	34
3.5.5.1	READ . . . . .	34
3.5.6	Endpunkt in der REST-API: /api/subjects/\$id/timetable . . . . .	35
3.5.6.1	READ . . . . .	35
3.6	Schnittstellen für Klassen . . . . .	37
3.6.1	Endpunkt in der REST-API: /api/classes/\$id . . . . .	37
3.6.1.1	READ . . . . .	37
3.6.2	Endpunkt in der REST-API: /api/classes/\$id/subjects . . . . .	38
3.6.2.1	READ . . . . .	38
3.6.3	Endpunkt in der REST-API: /api/classes/\$id/students . . . . .	39
3.6.3.1	READ . . . . .	39
3.6.4	Endpunkt in der REST-API: /api/classes/\$id/teachers . . . . .	40
3.6.4.1	READ . . . . .	40
3.6.5	Endpunkt in der REST-API: /api/classes/\$id/representatives . . . . .	41
3.6.5.1	READ . . . . .	42
<b>4</b>	<b>Weiterführende Konzepte</b>	<b>44</b>
4.1	Löschkonzept . . . . .	44
4.2	Glossar . . . . .	44
4.2.1	Begriffe der Fachdomäne . . . . .	44
4.2.2	Technische Begriffe . . . . .	45
	<b>Abbildungsverzeichnis</b>	<b>47</b>
	<b>Tabellenverzeichnis</b>	<b>48</b>
	<b>Listings</b>	<b>50</b>

## 0.1 Datenmodell-Beispiele

### 0.1.1 Benutzer (Datenmodelle)

```
1 {
2   id: "USER-01",
3   student-id: "12345",
4   name: "Leming",
5   surname: "Zobel",
6   birtdate: "2003-01-03",
7   sex: "male",
8   memberships: [
9     {
10      school: "SCHULE-01",
11      role: "student",
12      start: "2009-09-01",
13      end: "2016-08-31",
14      school-years: ["SJ-09/10", "SJ-10/11", "SJ-11/12", "SJ-13/14", "SJ-14/15", "SJ-15/16"]
15    }, {
16      school: "SCHULE-04",
17      role: "student",
18      start: "2016-09-01",
19      school-years: ["SJ-16/17", "SJ-17/18", "SJ-18/19", "SJ-19/20", "SJ-20/21"]
20    }, {
21      school: "SCHULE-02",
22      role: "external-student",
23      start: "2019-09-01",
24      end: "2020-08-31",
25      school-years: ["SJ-19/20"]
26    },
27  ],
28  guardians: [
29    {
30      user: "USER-02",
31      start: "2009-09-01",
32      end: "2020-01-03",
33    }, {
34      user: "USER-04",
35      start: "2009-09-01",
36      end: "2020-01-03",
37    }
38  ],
39  classes: [
40    {
41      class: "KLASSE-0001",
42      school: "SCHULE-01",
43      school-year: "SJ-09/10",
44      start: "2009-09-01",
45      end: "2010-08-31",
46    }, {
47      class: "KLASSE-0002",
48      school: "SCHULE-01",
```

```

49     school-year: "SJ-10/11",
50     start: "2010-09-01",
51     end: "2011-08-31",
52 }, {
53     class: "KLASSE-0003",
54     school: "SCHULE-01",
55     school-year: "SJ-10/11",
56     start: "2010-09-01",
57     end: "2011-08-31",
58 },
59 ],
60 subjects: [
61     {
62         subject_id: "SUBJECT-0001",
63         subject_ref_id: "DE",
64         school: "SCHULE-01",
65         school-year: "SJ-09/10",
66         start: "2009-09-01",
67         end: "2010-02-28".
68         time_tabel [
69             {
70                 day: "1",
71                 start: "08:00:00",
72                 end: "08:45:00",
73                 repeat: "weekly"
74             }, {
75                 day: "2",
76                 start: "08:00:00",
77                 end: "08:45:00",
78                 repeat: "weekly"
79             }, {
80                 day: "3",
81                 start: "08:50:00",
82                 end: "09:35:00",
83                 repeat: "biweekly",
84                 start: "week-1"
85             }, {
86                 day: "4",
87                 start: "08:50:00",
88                 end: "09:35:00",
89                 repeat: "biweekly",
90                 start: "week-2"
91             }, {
92                 day: "3",
93                 start: "08:50:00",
94                 end: "09:35:00",
95                 repeat: "once",
96                 date: "2009-10-30"
97             }
98         ], {
99             subject_id: "SUBJECT-0002",
100             subject_ref_id: "MA",
101             school: "SCHULE-01",

```

```

102     school-year: "SJ-09/10",
103     start: "2009-09-01",
104     end: "2010-08-31"
105 },
106
107 ]
108 }

```

Listing 1: Beispiel Benutzer mit Rolle 'student'

```

1  {
2    id: "USER-02",
3    name: "Altes Leming 1",
4    surname: "Zobel",
5    birtdate: "2003-01-03",
6    sex: "female",
7    memberships: [
8      {
9        school: "SCHULE-01",
10       role: "guardian",
11       start: "2009-09-01",
12       end: "2016-08-31",
13       school-years: ["SJ-09/10", "SJ-10/11", "SJ-11/12", "SJ-13/14", "SJ-14/15", "SJ-15/16"]
14     }, {
15       school: "SCHULE-04",
16       role: "guardian",
17       start: "2016-09-01",
18       school-years: ["SJ-16/17", "SJ-17/18", "SJ-18/19", "SJ-19/20", "SJ-20/21"]
19     }, {
20       school: "SCHULE-02",
21       role: "guardian",
22       start: "2019-09-01",
23       end: "2020-08-31",
24       school-years: ["SJ-19/20"]
25     }, {
26       school: "SCHULE-02",
27       role: "teacher",
28       start: "2019-09-01"
29     }
30   ],
31   children: [
32     {
33       user: "USER-01",
34       start: "2009-09-01",
35       end: "2020-01-03",
36     }, {
37       user: "USER-03",
38       start: "2009-09-01",
39       end: "2020-01-03",
40     }
41   ],
42   classes: [
43     {

```

```

44     class: "KLASSE-0031",
45     school: "SCHULE-02",
46     school-year: "SJ-09/10",
47     start: "2009-09-01",
48     end: "2010-08-31",
49   }, {
50     class: "KLASSE-0032",
51     school: "SCHULE-02",
52     school-year: "SJ-20/21",
53     start: "2020-09-01",
54     end: "2021-08-31",
55   }, {
56     class: "KLASSE-0033",
57     school: "SCHULE-02",
58     school-year: "SJ-20/21",
59     start: "2020-09-01",
60     end: "2021-08-31",
61   },
62 ]
63 }

```

Listing 2: Beispiel für Benutzer mit Rollen 'teacher' und 'guardian'

```

1 {
2   subject: "SUBJECT-0001",
3   name: "Deutsch 1-A",
4   school-subject: ["DE",],
5   school: "SCHULE-01",
6   school-year: "SJ-09/10",
7   start: "2009-09-01",
8   end: "2010-02-28",
9   classes: [ "KLASSE-01", "KLASSE-03", "KLASSE-05" ],
10  grade: [ "1" ],
11  students: [
12    {
13      user: "USER-01",
14      start: "2009-09-01",
15      end: "2010-02-28",
16    }, {
17      user: "USER-06",
18      start: "2009-09-01",
19      end: "2010-02-28",
20    }, {
21      user: "USER-07",
22      start: "2009-09-01",
23      end: "2009-12-31",
24    },
25  ],
26  teachers: [
27    {
28      user: "USER-08",
29      start: "2009-09-01",
30      end: "2010-02-28",
31    }, {

```

```

32     user: "USER-09",
33     start: "2009-09-01",
34     end: "2009-12-31",
35 }, {
36     user: "USER-10",
37     start: "2009-10-05",
38     end: "2009-10-05",
39 },
40 ],
41 timetable [
42     {
43         day: "1",
44         start: "08:00:00",
45         end: "08:45:00",
46         repeat: "weekly"
47     }, {
48         day: "2",
49         start: "08:00:00",
50         end: "08:45:00",
51         repeat: "weekly"
52     }, {
53         day: "3",
54         start: "08:50:00",
55         end: "09:35:00",
56         repeat: "biweekly",
57         week: "week-1"
58     }, {
59         day: "4",
60         start: "08:50:00",
61         end: "09:35:00",
62         repeat: "biweekly",
63         week: "week-2"
64     }, {
65         day: "3",
66         start: "08:50:00",
67         end: "09:35:00",
68         repeat: "onetime",
69         date: "2009-10-30"
70     }
71 ]
72 }

```

Listing 3: Beispiel eines Schulfachs

```

1 {
2   class: "KLASSE-01",
3   name: "Klasse 1-A",
4   school: "SCHULE-01",
5   school-year: "SJ-09/10",
6   start: "2009-09-01",
7   end: "2010-08-31",
8   grade: [ "1", ],
9   subjects: [ "SUBJECT-0001", "SUBJECT-0003", "SUBJECT-0004", ],
10  students: [

```

```

11  {
12    user: "USER-01",
13  }, {
14    user: "USER-06",
15  }, {
16    user: "USER-07",
17    end: "2009-12-31",
18  }, {
19    user: "USER-10",
20  }, {
21    user: "USER-11",
22  },
23 ],
24 teachers: [
25   {
26     user: "USER-208",
27     order: [
28       {
29         order: "1",
30       },
31     ],
32   }, {
33     user: "USER-209",
34     order: [
35       {
36         order: "2",
37       },
38     ],
39   },
40 ],
41 representatives: [
42   {
43     user: "USER-10",
44     role: "student",
45     order: "1",
46   }, {
47     user: "USER-07",
48     end: "2009-12-31",
49     role: "student",
50     order: "2",
51   }, {
52     user: "USER-01",
53     start: "2010-01-04",
54     role: "student",
55     order: "2",
56   }, {
57     user: "USER-114",
58     role: "guardian",
59     order: "1",
60   }, {
61     user: "USER-115",
62     role: "guardian",
63     order: "2",

```



```

64     },
65   ]
66 }

```

Listing 4: Klassen-Datenmodell Beispiel 2: Grundschulklasse

```

1  {
2    class: "KLASSE-11",
3    name: "Jarganstuffe 11",
4    school: "SCHULE-04",
5    school-year: "SJ-20/21",
6    start: "2020-09-01",
7    end: "2021-08-31",
8    grade: [ "11", ],
9    subjects: [ ],
10   students: [
11     {
12       user: "USER-01",
13     }, {
14       user: "USER-30",
15     }, {
16       user: "USER-31",
17     }, {
18       user: "USER-32",
19     }, {
20       user: "USER-33",
21     },
22   ],
23   teachers: [
24     {
25       user: "USER-228",
26       end: "2021-01-14",
27       order: [
28         {
29           order: "1",
30           end: "2021-01-14",
31         },
32       ],
33     }, {
34       user: "USER-229",
35       order: [
36         {
37           order: "2",
38           end: "2021-01-14",
39         }, {
40           order: "1",
41           start: "2021-01-15",
42         },
43       ], {
44       user: "USER-230",
45       start: "2021-01-15",
46       order: [
47         {
48           order: "2",

```

```

49     start: "2021-01-15",
50   },
51 ],
52 },
53 ],
54 representatives: [
55   {
56     user: "USER-01",
57     role: "student",
58     order: "1",
59   }, {
60     user: "USER-31",
61     role: "student",
62     order: "2",
63   }, {
64     user: "USER-141",
65     start: "2020-09-10",
66     role: "guardian",
67     order: "1",
68   }, {
69     user: "USER-142",
70     start: "2020-09-10",
71     role: "guardian",
72     order: "2",
73   },
74 ]
75 }

```

Listing 5: Klassen-Datenmodell Beispiel 2: Jahrgangstufe 11, Sekundarstufe 2

```

1 {
2   class: "KLASSE-21",
3   name: "Klasse 3,4 A",
4   school: "SCHULE-07",
5   school-year: "SJ-15/16",
6   start: "2015-09-01",
7   end: "2016-08-31",
8   grade: [ "3", "4", ],
9   subjects: [ "SUBJECT-0101", "SUBJECT-0102", "SUBJECT-0103", ],
10  students: [
11    {
12      user: "USER-51",
13    }, {
14      user: "USER-52",
15    }, {
16      user: "USER-53",
17    }, {
18      user: "USER-55",
19    }, {
20      user: "USER-54",
21      start: "2016-03-01",
22    },
23  ],
24  teachers: [

```

```

25 {
26   user: "USER-256",
27   order: [
28     {
29       order: "1",
30     },
31   ],
32 }, {
33   user: "USER-257",
34   order: [
35     {
36       order: "1",
37     },
38   ],
39 }, {
40   user: "USER-258",
41   order: [
42     {
43       order: "2",
44     },
45   ],
46 },
47 ],
48 representatives: [
49   {
50     user: "USER-55",
51     role: "student",
52     order: "1",
53   }, {
54     user: "USER-52",
55     role: "student",
56     order: "2",
57   }, {
58     user: "USER-158",
59     start: "2015-09-10",
60     role: "guardian",
61     order: "1",
62   }, {
63     user: "USER-159",
64     start: "2015-09-10",
65     role: "guardian",
66     order: "2",
67   },
68 ]
69 }

```

Listing 6: Klassen-Datenmodell Beispiel 3: Jahrgangsübergreifende Klasse

```

1 {
2   school: "SCHULE-01",
3   name: "Zobel Grundschule",
4   classes: [
5     {
6       class: "KLASSE-0001",

```

```

7     start: "2020-09-01",
8     end: "2021-08-31",
9 }, {
10    class: "KLASSE-0002",
11    start: "2010-09-01",
12    end: "2011-08-31",
13 }, {
14    class: "KLASSE-0003",
15    start: "2010-09-01",
16    end: "2011-08-31",
17 },
18 ],
19 users: [
20     {
21         user: "USER-02",
22         role: "guardian",
23         start: "2009-09-01",
24         end: "2016-08-31",
25         school-years: ["SJ-09/10", "SJ-10/11", "SJ-11/12", "SJ-13/14", "SJ-14/15",
26                        "SJ-15/16"],
27     }, {
28         user: "USER-01",
29         role: "student",
30         start: "2009-09-01",
31         end: "2016-08-31",
32         school-years: ["SJ-09/10", "SJ-10/11", "SJ-11/12", "SJ-13/14", "SJ-14/15",
33                        "SJ-15/16"],
34     }, {
35         user: "USER-208",
36         role: "teacher",
37         start: "2001-09-01",
38     }, {
39         user: "USER-209",
40         role: "teacher",
41         start: "2007-09-01",
42     },
43 ],
44 subjects: [
45     {
46         subject: "SUBJECT-0001",
47         start: "2009-09-01",
48         end: "2010-02-28",
49     }, {
50         subject: "SUBJECT-0101",
51         start: "2009-09-01",
52         end: "2010-02-28",
53     },
54 ],
55 }

```

Listing 7: Schulen-Datenmodell Beispiel 1

# 1 Einleitung und Vorüberlegungen

Ziel dieses Dokument ist es, eine REST-API und die darauf verwendeten Datenobjekte zu spezifizieren bzw. zu modellieren, um Daten aus einem IDM-System beziehen zu können.

Der IDM-Provider ist verpflichtet, mindestens die geforderten Schnittstellen und Protokolle mit der angegebenen Verfügbarkeitsanforderung bereitzustellen. Im Gegenzug erhält der IDM-Provider über viele Objekte die Hoheit der zentralen ID-Vergabe. Eine zentralisierte ID-Vergabe ist notwendig, damit verschiedene nachgelagerte Systeme (z.B. Schulverwaltungssoftwares, Lernplattformen) interoperabel sind.

Es werden die notwendigen Endpunkte definiert, die jeweils zugelassenen Operationen nebst verwendeter HTTP-Methode festgelegt, die Datenobjekte in JSON modelliert sowie der Workflow darauf dargestellt.

## 1.1 Benutzerrollen

An dieser Stelle listen wir eine Reihe von im Kontext “Schule“ bekannten Rollen auf, die auf IDM-Provider-Seite definiert und im Client für die Berechtigungssteuerung genutzt werden können.

Bei Bedarf können IDM-Provider und Client weitere Rollen festlegen, die additiv zu vergeben und nicht Teil dieser Konzeption sind.

<b>Rollenname</b>	<b>Beschreibung der Rolle</b>
guest	Gäste und nicht authentifizierte Benutzer
user	authentifizierte Benutzer
student	Schüler
external-student	Schüler von anderen Schulen, die nur einzelne Fächer oder Kurse besuchen
guardian	Eltern, Erziehungsberechtigte, Vormünder von Schülern
teacher	Lehrkräfte
external-expert	Externe Experten zur Unterstützung des Unterrichts (auch Schulassistenten und pädagogische Mitarbeiter)
principal	Schulleitung
school-admin	Schul-Administrator
school-board	Schulträger
fed-school-board	Mitarbeiter/-in des Schulministeriums
sync-system	Systeme, welchen ein Sync mit allen Daten erlaubt ist

Tabelle 1.1: Mögliche Benutzerrollen im Kontext “Schule“

## 2 Autorisierung und Authentifizierung

Um den Zugriff auf Client und REST-API gleichermaßen zu steuern und dabei im Client keine Anmeldeinformationen vorhalten zu müssen, wird eine Autorisierung und Authentifizierung per *OAuth2-Protokoll* und *OpenID Connect* vorgeschrieben.

### 2.1 OAuth2

OAuth2 ist ein Standardprotokoll für die Benutzerautorisierung [?].

Der Autorisierungsserver des IDM-Providers muss gemäß RFC 6749 folgende *Endpoints* bereitstellen:

Endpoint	Funktion
Authorization	Initiierung der Autorisierung und Benutzerzustimmung durch parametrisierten Aufruf
Token	Liefert gegen Authorization Code Access Token zurück

Tabelle 2.1: Endpunkte, die durch den IDM-Provider für die Anmeldung per OAuth2 zur Verfügung gestellt werden müssen

OAuth2 definiert sogenannte *Grant Types*, die vom *Client* über das Setzen eines oder mehrerer *response\_type* beim Aufruf des *Authorization Endpoint* gewählt werden können. Diese *Grant Types* beschreiben Möglichkeiten, wie ein Client einen *Access Token* erlangt, über den im Namen des Benutzers anschließend eine API aufgerufen werden kann. Während *response\_type=code* den *Authorization Code Grant* initiiert, liefert *response\_type=token* den Access Token direkt nach Autorisierung zurück. Zudem lassen sich über *Access Token Scopes* vom Client Anwendungsbereiche des vom Autorisierungsserver gelieferten Access Tokens anfordern [?, Abschnitt 3.3].

Die *Internet Engineering Task Force (IETF)* empfiehlt die Verwendung des *Authorization Code Grant* [?, Unterabschnitt 2.1.1]. Je nachdem, ob es sich um einen *Confidential* oder *Public Client* handelt, d.h., ob ein *Client Secret* verwendet werden kann, wird ein *Proof Key for Code Exchange (PKCE)* empfohlen bzw. vorgeschrieben. Der PKCE ist eine Erweiterung des *Authorization Code Grant*, um Cross-Site-Request-Forgery (CSRF) oder Authentication-Code-Injection-Angriffe zu verhindern [?].

Der Ablauf der Autorisierung per OAuth2-Protokoll im *Authorization Code Grant* ist in Abbildung *Abbildung 2.1* dargestellt. Dieser wird durch den *Client* gestartet, gefolgt

von der Autorisierung durch den *Resource Owners* beim Autorisierungsserver. Der Autorisierungsserver sendet einen *Authorization Code* an den *Client*, der diesen wiederum direkt beim Autorisierungsserver gegen den *Access Token* tauscht. Über den Access Token erhält der Benutzer schließlich Zugriff auf die REST-API.

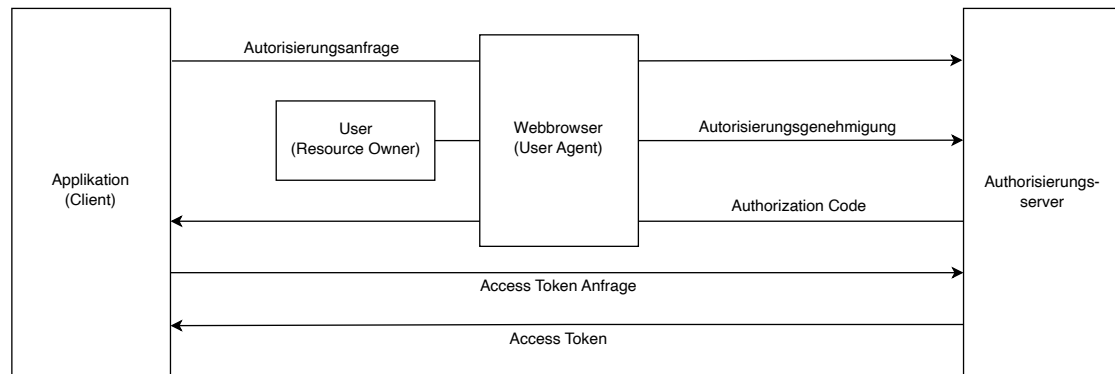


Abbildung 2.1: Ablauf des Authentication Code Grant

## 2.2 OpenID Connect

OpenID Connect ist eine Spezifikation, wie *ID Tokens* mit personenbezogenen Daten ausgegeben werden [?]. Dabei setzt OpenID Connect auf das OAuth2-Protokoll auf. OpenID Connect definiert ein weiteres *response\_type=id\_token*. Dieser ermöglicht es, neben dem Access Token den sogenannten *ID Token* in Form eines JSON Web Tokens [?] vom Autorisierungsserver anzufordern. Um OpenID Connect nutzen zu können, muss im in *Abschnitt 2.1: OAuth2* beschriebenen Autorisierungsprozess der Parameter *Scopes* um *openid* erweitert werden.

Über den ID Token erlangt der Client Zugriff auf personenbezogene Daten des Benutzers. *OpenID Connect* definiert eine Reihe von Standard-*claims* [?, Abschnitt StandardClaims]), die im *Scope*-Parameter beim Aufruf der Autorisierungs-URL aufgelistet werden. Standard-*claims* können um Custom-*claims* erweitert werden, um die gewünschten Benutzerinformationen im ID Token anfordern.

## 2.3 Einschränkung der Daten

Die Spezifikation der REST-API sieht vor, dass die Sichtbarkeit der Daten an den definierten Endpunkte bereits im JSON-Objekt gemäß der Berechtigung des anfragenden Benutzers berücksichtigt ist. Daher muss der *ID Token* Informationen enthalten, über die der IDM-Provider den Benutzer eindeutig identifizieren kann. Die Informationen aus dem ID Token reichen ggf. jedoch nicht aus, um die Sichtbarkeit der Daten korrekt zu



beschränken, da einem Benutzer im IDM mehrere Schule-Rolle-Kombinationen zugewiesen sein können (z.B. Tätigkeit als Lehrkraft an verschiedenen Schulen oder sowohl Lehrkraft als auch Elternteil eines Schülers an derselben Schule). Daher muss die Information, mit welcher Kombination von Schule und Rolle innerhalb dieser Schule der Aufruf eines REST-API-Endpunkts durchgeführt wird, an den IDM-Provider übermittelt werden. Dies geschieht durch die Erweiterung der *Scopes*-Liste um den Scope für die Schule und die Rolle als Parameterübergabe beim Autorisierungsvorgang. *Listing 2.1* zeigt einen exemplarischen Aufruf des *Authorization Endpoints* mit Übergabe der *Scopes*. Die zusätzlichen *Scopes* für Schule und Rolle sind somit im Access Token hinterlegt und können bei der Generierung des zurückzuliefernden JSON-Objekts verwendet werden.

```
1 https://<URL zum Authorization Endpoint>?response_type=code
2 &client_id=<Identifizier von Client-App>
3 &redirect_uri=<Redirect-URL>
4 &scope=openid teacher school
5 &state=<Undurchschaubarer Wert fuer Sicherheitszwecke>
```

Listing 2.1: Beispielhafter Aufruf des Authorization Endpoints

Ein Sonderfall stellen dabei Benutzerkonten mit der Rolle "sync-system" dar. Diese haben bei Aufruf eines REST-API-Endpunkts grundsätzlich keine Beschränkung auf eine einzelne Schule. Daher muss beim Autorisierungsvorgang auch kein *Scope* für die Schule an den Autorisierungsserver übergeben werden.

Sofern für ein Benutzerkonto im IDM mehrere Schule-Rolle-Kombinationen hinterlegt sein können, müssen Schule und Rolle bei Start des Authentifizierungsprozesses gegebenenfalls durch den Benutzer wählbar sein. Für einen Schule-Rolle-Wechsel ist eine Neuansmeldung bzw. Aktualisierung des Access Tokens bezüglich der *Scopes* für Schule und Rolle notwendig.

## 3 REST-API-Definition

In den folgenden Abschnitten werden die einzelnen Endpunkte der REST-API definiert. Jeder Endpunkt stellt einen Zugriffspunkt auf einen Datentyp dar und es wird definiert, inwiefern die einzelnen Operationen CREATE, READ, UPDATE und DELETE zugelassen sind und welche HTTP-Methode dabei Verwendung findet. Auch erfolgt jeweils die Definition der JSON-Objekte.

### 3.1 Schnittstellen für Schulfächer

Die Schnittstelle für Schulfächer listet die Teilmenge der abgestimmten Referenzschulfächer auf, welche von dem IDM unterstützt werden. Die Gesamtmenge aller abgestimmten Referenzschulfächer kann *Tabelle 3.1* entnommen werden.

ID	Abkürzung	Ausgeschriebener Name

Tabelle 3.1: Liste der abgestimmten Referenzschulfächer

#### 3.1.1 Endpunkt in der REST-API: /api/school-subjects

Die *Tabelle 3.2* listet auf, welche Operationen zugelassen sind und welche HTTP-Methoden dabei verwendet werden. In der *Tabelle 3.1* ist eine Liste der vollständigen Fächer enthalten.

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

Tabelle 3.2: Zugelassene Operationen auf /api/school-subjects

##### 3.1.1.1 READ

Es sind nur Anfragen mit der HTTP-GET-Methode für ein READ auf die Daten zugelassen. Bei einer lesenden Anfrage wird eine Liste der Teilmenge der Referenzschulfächer, welche vom IDM unterstützt werden, zurückgegeben.

Die Antwort erfolgt mit dem JSON-Objekt *Listing 3.1*. Die einzelnen Felder der Antwort werden in *Tabelle 3.3* beschrieben.

```

1 [
2 {
3   school-subject: "<STRING>",
4   short-name: "<STRING>",
5   name: "<STRING>"
6 },
7 ...
8 ]

```

Listing 3.1: JSON-Antwort für einen GET-Aufruf des Pfads `/api/school-subjects`

Feldname	Datentyp	Beschreibung
school-subject	STRING	Eine eindeutige Zeichenkette, die vom IDM-Provider vergeben wird. Sie darf nur aus alphanumerischen Zeichen und Bindestrich bestehen.
name	STRING	Der ausgeschriebene Name eines Schulfaches

Tabelle 3.3: Beschreibung der Felder in einem JSON-Objekt für ein Schulfach

## 3.2 Schnittstellen für Schuljahre

Die Schnittstelle für Schuljahre hat die Aufgabe alle im IDM verfügbaren Schuljahre mit ihren allgemeinen Informationen bereitzustellen.

### 3.2.1 Endpunkt in der REST-API: `/api/school-years`

Die *Tabelle 3.4* listet auf, welche Operationen zugelassen sind und welche HTTP-Methoden dabei verwendet werden.

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

Tabelle 3.4: Zugelassene Operationen auf `/api/school-years`

#### 3.2.1.1 READ

Es sind nur Anfragen mit der HTTP-GET-Methode für ein READ auf die Daten zugelassen. Unter diesem Pfad wird eine Liste mit allen im System existierenden Schuljahren

zurückgegeben. Jedes Schuljahr hat einen eindeutigen Identifikator, einen Zeitraum, von wann bis wann das Schuljahr andauerte, und einen Anzeigenamen.

Die Antwort erfolgt mit dem JSON-Objekt *Listing 3.2*. Die einzelnen Felder der Antwort werden in *Tabelle 3.5* beschrieben.

```

1  [
2  {
3    school-year: "<STRING>",
4    start: "<DATE>",
5    end: "<DATE>",
6    name: "<STRING>",
7  },
8  ...
9  ]

```

Listing 3.2: JSON-Antwort für einen GET-Aufruf des Pfads `/api/school-years`

Feldname	Datentyp	Beschreibung
school-year	STRING	Eine eindeutige Zeichenkette, welche das Schuljahr im IDM identifiziert.
start	DATE	Ein Datumsangabe nach ISO-8601 in dem Format YYYY-MM-DD, welche angibt, wann das Schuljahr begonnen hat.
ende	DATE	Ein Datumsangabe nach ISO-8601 in dem Format YYYY-MM-DD, welche angibt, wann das Schuljahr endet.
name	STRING	Eine Name für das Schuljahr wie z.b.: 2020-2021

Tabelle 3.5: Beschreibung der Felder in einem JSON-Objekt aus der Liste der Schuljahre

## 3.3 Schnittstellen für Schulen

### 3.3.1 Endpunkt in der REST-API: `/api/schools`

Die *Tabelle 3.6* listet auf, welche Operationen zugelassen sind und welche HTTP-Methoden dabei verwendet werden.

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

Tabelle 3.6: Zugelassene Operationen auf `/api/schools`

### 3.3.1.1 READ

Es sind nur Anfragen mit der HTTP-GET-Methode für ein READ auf die Daten zugelassen. Der Endpunkt gibt eine Liste mit allen Schul-IDs aus, die es im System gibt und für die Daten unter `/api/schools/$id` abgerufen werden können.

Die Antwort erfolgt mit dem JSON-Objekt *Listing 3.3*. Die einzelnen Felder der Antwort werden in *Tabelle 3.7* beschrieben.

```
1 [
2   "<STRING>",
3   ...
4 ]
```

Listing 3.3: JSON-Antwort für einen GET-Aufruf des Pfads `/api/schools`

Feldname	Datentyp	Beschreibung
	STRING	Eindeutiger Identifikations-String der Schule im IDM, mit dem unter <code>/api/schools/\$id</code> die Schule abgerufen werden kann.

Tabelle 3.7: Beschreibung des Felder in einem JSON-Objekt, was die Schul-IDs wiedergibt

### 3.3.2 Endpunkt in der REST-API: `/api/schools/$id`

Die *Tabelle 3.8* listet auf, welche Operationen zugelassen sind und welche HTTP-Methoden dabei verwendet werden.

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

Tabelle 3.8: Zugelassene Operationen auf `/api/schools/$id`

#### 3.3.2.1 READ

Es sind nur Anfragen mit der HTTP-GET-Methode für ein READ auf die Daten zugelassen. Diese Route gibt für die per ID ausgewählte Schule die Stammdaten zurück.

Die Antwort erfolgt mit dem JSON-Objekt *Listing 3.4*. Die einzelnen Felder der Antwort werden in *Tabelle 3.9* beschrieben.

```
1 {
2   school: "<STRING>",
3   name: "<STRING>",
```

4 } }

Listing 3.4: JSON-Antwort für einen GET-Aufruf des Pfads `/api/school/$id`

Feldname	Datentyp	Beschreibung
school	STRING	Eindeutiger Identifikations-String der Schule im IDM
name	STRING	Name der Schule

Tabelle 3.9: Beschreibung der Felder in einem JSON-Objekt für das Zuordnen eines Benutzer in einer Rolle zu einer Schule

### 3.3.3 Endpunkt in der REST-API: `/api/schools/$id/users`

Die *Tabelle 3.10* listet auf, welche Operationen zugelassen sind und welche HTTP-Methoden dabei verwendet werden.

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

Tabelle 3.10: Zugelassene Operationen auf `/api/schools/$id/users`

#### 3.3.3.1 READ

Es sind nur Anfragen mit der HTTP-GET-Methode für ein READ auf die Daten zugelassen. Dieser Endpunkt listet die Informationen über Benutzer und ihre Rollen an der Schule auf. Für jede Kombination aus Benutzer, Rolle und Startdatum gibt es einen Eintrag in der Liste, die zurückgegeben wird. Bei manchen Kombinationen gibt es zudem noch ein Datum, bis zu dem der Benutzer die Rolle innehatte. Zudem kann es bei einigen Rollen noch die Zusatzinformation geben, in welche Schuljahren der User die Rolle inne hatte.

Die Antwort erfolgt mit dem JSON-Objekt *Listing 3.5*. Die einzelnen Felder der Antwort werden in *Tabelle 3.11* beschrieben.

```

1 [
2   {
3     user: "<STRING>",
4     role: "<STRING>",
5     start: "<DATE>",
6     end: "<DATE>",
7     school-years: ["<STRING>", ...],
8   },

```

```

9 | ...
10 | ]

```

Listing 3.5: JSON-Antwort für einen GET-Aufruf des Pfads `/api/schools/$id/users`

Feldname	Datentyp	Beschreibung
user	STRING	Zeichenkette, welche eine ID von einem Benutzer ist, der unter <code>/api/subjects/\$id</code> abgerufen werden kann.
role	STRING	Rolle, die der Benutzer innehat.
start	DATE	Datum, ab dem der Benutzer die Rolle innehat
end	DATE	Optional; Gibt an, bis zu welchen Tag der Benutzer die Rolle innehatte.
school-years	Liste	Optional; Liste von Zeichenketten, welche Referenzen auf Schuljahre sind, die unter <code>/api/school-years/\$id</code> abgerufen werden können.

Tabelle 3.11: Beschreibung der Felder in einem JSON-Objekt für das Zuordnen eines Benutzer in einer Rolle zu einer Schule

### 3.3.4 Endpunkt in der REST-API: `/api/schools/$id/classes`

Die *Tabelle 3.12* listet auf, welche Operationen zugelassen sind und welche HTTP-Methoden dabei verwendet werden.

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

Tabelle 3.12: Zugelassene Operationen auf `/api/schools/$id/classes`

#### 3.3.4.1 READ

Es sind nur Anfragen mit der HTTP-GET-Methode für ein READ auf die Daten zugelassen. Der Endpunkt gibt eine Liste von Referenz-IDs für Klassen an der Schule zurück, welche unter `/api/classes/$id` abgerufen werden können.

Die Antwort erfolgt mit dem JSON-Objekt *Listing 3.6*. Die einzelnen Felder der Antwort werden in *Tabelle 3.13* beschrieben.

```

1 | [
2 |   "<STRING>",
3 |   ...
4 | ]

```

---

Listing 3.6: JSON-Antwort für einen GET-Aufruf des Pfads `/api/schools/$id/classes`

Feldname	Datentyp	Beschreibung
	STRING	Zeichenkette, welche eine ID einer Klasse ist, die unter <code>/api/classes/\$id</code> abgerufen werden kann.

Tabelle 3.13: Beschreibung der Felder in einem JSON-Objekt, welches die Klassen an einer Schule repräsentiert.

### 3.3.5 Endpunkt in der REST-API: `/api/schools/$id/subjects`

Die *Tabelle 3.14* listet auf, welche Operationen zugelassen sind und welche HTTP-Methoden dabei verwendet werden.

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

Tabelle 3.14: Zugelassene Operationen auf `/api/schools/$id/subjects`

#### 3.3.5.1 READ

Es sind nur Anfragen mit der HTTP-GET-Methode für ein READ auf die Daten zugelassen. Der Endpunkt gibt eine Liste von Referenz-IDs für Unterrichtsfächer zurück, welche unter `/api/subjects/$id` abgerufen werden können.

Die Antwort erfolgt mit dem JSON-Objekt *Listing 3.7*. Die einzelnen Felder der Antwort werden in *Tabelle 3.15* beschrieben.

```
1 [
2   "<STRING>" ,
3   ...
4 ]
```

Listing 3.7: JSON-Antwort für einen GET-Aufruf des Pfads `/api/schools/$id/subjects`

Feldname	Datentyp	Beschreibung
	STRING	Zeichenkette, welche eine ID von einem Unterrichtsfach ist, das unter <code>/api/subjects/\$id</code> abgerufen werden kann.

Tabelle 3.15: Beschreibung der Felder in einem JSON-Objekt, welches die Unterrichtsfächer an einer Schule repräsentieren



## 3.4 Schnittstellen für Benutzer

### 3.4.1 Endpunkt in der REST-API: /api/users

Die *Tabelle 3.16* listet auf, welche Operationen zugelassen sind und welche HTTP-Methoden dabei verwendet werden.

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

Tabelle 3.16: Zugelassene Operationen auf /api/users

#### 3.4.1.1 READ

Es sind nur Anfragen mit der HTTP-GET-Methode für ein READ auf die Daten zugelassen. Die Route gibt für den eingeloggtten Benutzer seine persönlichen Daten wieder. Die Antwort erfolgt mit dem JSON-Objekt *Listing 3.8*. Die einzelnen Felder der Antwort werden in *Tabelle 3.17* beschrieben.

```
1 {  
2   id: "<STRING>",  
3   student-id: "<STRING>",  
4   name: "<STRING>",  
5   surname: "<STRING>",  
6   dateofbirth: "<DATE>",  
7   sex: "<ENUM>",  
8 }
```

Listing 3.8: JSON-Antwort für einen GET-Aufruf des Pfads /api/users

Feldname	Datentyp	Beschreibung
id	STRING	ID des Benutzers, mit der unter /api/users/\$id weitere Informationen abgefragt werden können.
student-id	STRING	Landesweit gültige Schüler-Identifikationsnummer
name	STRING	Vorname des Benutzers
surname	STRING	Nachname des Benutzers
dateofbirth	DATE	Geburtstag des Benutzers

Tabelle 3.17: Beschreibung der Felder in einem JSON-Objekt von einem Benutzer

Feldname	Datentyp	Beschreibung
sex	ENUM	Geschlecht des Benutzers <ul style="list-style-type: none"> <li>• divers steht für divers</li> <li>• female steht für weiblich</li> <li>• male steht für männlich</li> </ul>

Tabelle 3.17: Beschreibung der Felder in einem JSON-Objekt von einem Benutzer

### 3.4.2 Endpunkt in der REST-API: `/api/users/$id`

Die *Tabelle 3.18* listet auf, welche Operationen zugelassen sind und welche HTTP-Methoden dabei verwendet werden.

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

Tabelle 3.18: Zugelassene Operationen auf `/api/users/$id`

#### 3.4.2.1 READ

Es sind nur Anfragen mit der HTTP-GET-Methode für ein READ auf die Daten zugelassen. Die Route gibt für den per ID ausgewählten Benutzer seine persönlichen Daten wieder. Diese können eingeschränkt werden durch den Kontext, in dem sich der anfragende Benutzer befinden.

Die Antwort erfolgt mit dem JSON-Objekt *Listing 3.9*. Die einzelnen Felder der Antwort werden in *Tabelle 3.19* beschrieben.

```

1 {
2   id: "<STRING>",
3   name: "<STRING>",
4   surname: "<STRING>",
5   dateofbirth: "<DATE>",
6   sex: "<ENUM>",
7 }
```

Listing 3.9: JSON-Antwort für einen GET-Aufruf des Pfads `/api/users/$id`

Feldname	Datentyp	Beschreibung
id	STRING	ID des Benutzers, mit der unter <code>/api/users/\$id</code> weitere Informationen abgefragt werden können.
name	STRING	Vorname des Benutzers
surname	STRING	Nachname des Benutzers
dateofbirth	DATE	Geburtstag des Benutzers
sex	ENUM	Geschlecht des Benutzers <ul style="list-style-type: none"> <li>• divers steht für divers</li> <li>• female steht für weiblich</li> <li>• male steht für männlich</li> </ul>

Tabelle 3.19: Beschreibung der Felder in einem JSON-Objekt von einem Benutzer

### 3.4.3 Endpunkt in der REST-API: `/api/users/$id/memberships`

Die *Tabelle 3.20* listet auf, welche Operationen zugelassen sind und welche HTTP-Methoden dabei verwendet werden.

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

Tabelle 3.20: Zugelassene Operationen auf `/api/users/$id/memberships`

#### 3.4.3.1 READ

Es sind nur Anfragen mit der HTTP-GET-Methode für ein READ auf die Daten zugelassen. Sie gibt für den per ID ausgewählten Benutzer seine Zuordnungen im System wieder. Für diese Daten gelten die Einschränkungen die durch den Kontext des anfragenden Benutzers gegeben sind.

Die Antwort erfolgt mit dem JSON-Objekt *Listing 3.10*. Die einzelnen Felder der Antwort werden in *Tabelle 3.21* beschrieben.

```

1 {
2   school: "<STRING>",
3   role: "<STRING>",
4   start: "<CALENDARDATE>",
5   end: "<CALENDARDATE>",
6   school-years: ["<STRING>", ...]
7 },

```

Listing 3.10: JSON-Antwort für einen GET-Aufruf des Pfads  
/api/users/\$id/assignments

Feldname	Datentyp	Beschreibung
school	STRING	Zeichenkette, welche eine ID von einer Schule ist, mit der diese unter /api/schools/\$id abgerufen werden kann.
role	STRING	Rolle, die der Benutzer innehat.
start	CALENDARDATE	Datum, ab dem der Benutzer die Rolle innehat.
end	CALENDARDATE	Optional; Gibt an, bis zu welchen Datum der Benutzer die Rolle innehatte.
school-years	Liste	Optional; Liste von Zeichenketten, welche Referenzen auf Schuljahre sind, die unter /api/school-years/\$id abgerufen werden können.

Tabelle 3.21: Beschreibung der Felder in einem JSON-Objekt für das Zuordnen eines Benutzer in einer Rolle zu einer Schule

### 3.4.4 Endpunkt in der REST-API: /api/users/\$id/classes

Die *Tabelle 3.22* listet auf, welche Operationen zugelassen sind und welche HTTP-Methoden dabei verwendet werden.

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

Tabelle 3.22: Zugelassene Operationen auf /api/users/\$id/classes

#### 3.4.4.1 READ

Es sind nur Anfragen mit der HTTP-GET-Methode für ein READ auf die Daten zugelassen. Die Route gibt für den per ID ausgewählten Benutzer seine für ihn hinterlegten Klassen im System wieder. Für diese Daten gelten die Einschränkungen die durch den Kontext des anfragenden Benutzers gegeben sind.

Die Antwort erfolgt mit dem JSON-Objekt *Listing 3.11*. Die einzelnen Felder der Antwort werden in *Tabelle 3.23* beschrieben.

```

2  {
3    class: "<STRING>",
4    school: "<STRING>",
5    school-year: "<STRING>",
6    start: "<DATE>",
7    end: "<DATE>",
8  },
9  ...
10 ]

```

Listing 3.11: JSON-Antwort für einen GET-Aufruf des Pfads `/api/users/$id/classes`

Feldname	Datentyp	Beschreibung
class	STRING	ID der Klasse im System.
school	STRING	ID, zu welcher Schule die Klasse gehört, die Schule kann damit unter <code>/api/schools/\$id</code> abgefragt werden.
school-year	STRING	Schuljahr, in welchem die Klasse existiert. Die Daten des Schuljahres können unter <code>/api/school-years/\$id</code> abgerufen werden.
start	DATE	Zeitpunkt, ab dem das Klassenobjekt existiert, falls dies abweichend zum Startzeitpunkt des Schuljahres ist.
end	DATE	Zeitpunkt, bis zu dem das Klassenobjekt existiert, falls dies abweichend zum Endzeitpunkt des Schuljahres ist.

Tabelle 3.23: Beschreibung der Felder in einem JSON-Objekt aus der Liste der Klassen eines Benutzers

### 3.4.5 Endpunkt in der REST-API: `/api/users/$id/subjects`

Die *Tabelle 3.24* listet auf, welche Operationen zugelassen sind und welche HTTP-Methoden dabei verwendet werden.

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

Tabelle 3.24: Zugelassene Operationen auf `/api/users/$id/subjects`

### 3.4.5.1 READ

Es sind nur Anfragen mit der HTTP-GET-Methode für ein READ auf die Daten zugelassen. Gibt für den per ID ausgewählten Benutzer im Kontext des anfragenden Benutzers eine Liste mit Schulfächern für den ausgewählten Benutzer zurück. Die Antwort erfolgt mit dem JSON-Objekt *Listing 3.12*. Die einzelnen Felder der Antwort werden in *Tabelle 3.25* beschrieben.

```
1 [
2   "<STRING>",
3   ...
4 ]
```

Listing 3.12: JSON-Antwort für einen GET-Aufruf des Pfads `/api/users/$id/subjects`

Feldname	Datentyp	Beschreibung
	STRING	Referenz auf das Unterrichtsfach; kann als ID im Pfad <code>/api/subjects/\$id</code> verwendet werden.

Tabelle 3.25: Beschreibung der Zeichenkette in der JSON-Liste für Unterrichtsfächer eines Benutzers

### 3.4.6 Endpunkt in der REST-API: `/api/users/$id/children`

Die *Tabelle 3.26* listet auf, welche Operationen zugelassen sind und welche HTTP-Methoden dabei verwendet werden.

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

Tabelle 3.26: Zugelassene Operationen auf `/api/users/$id/children`

#### 3.4.6.1 READ

Es sind nur Anfragen mit der HTTP-GET-Methode für ein READ auf die Daten zugelassen. Die Route gibt für den per ID ausgewählten Benutzer Benutzer die Liste mit den IDs seiner Kinder wieder. Für diese Daten gelten die Einschränkungen die durch den Kontext des anfragenden Benutzers gegeben sind.

Die Antwort erfolgt mit dem JSON-Objekt *Listing 3.13*. Die einzelnen Felder der Antwort werden in *Tabelle 3.27* beschrieben.

```
1 [
2   "<STRING>",
```

```

3 | ...
4 | ]

```

Listing 3.13: JSON-Antwort für einen GET-Aufruf des Pfads `/api/users/$id/children`

Feldname	Datentyp	Beschreibung
	STRING	ID des Kindes, mit der unter <code>/api/users/\$id</code> weitere Informationen abgefragt werden können.

Tabelle 3.27: Beschreibung der Zeichenkette in der JSON-Liste für Kinder von Erziehungsberechtigten.

### 3.4.7 Endpunkt in der REST-API: `/api/users/$id/guardians`

Die *Tabelle 3.28* listet auf, welche Operationen zugelassen sind und welche HTTP-Methoden dabei verwendet werden.

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

Tabelle 3.28: Zugelassene Operationen auf `/api/users/$id/guardians`

#### 3.4.7.1 READ

Es sind nur Anfragen mit der HTTP-GET-Methode für ein READ auf die Daten zugelassen. Die Route gibt für den per ID ausgewählten Benutzer die Liste mit den IDs seiner Eltern, Erziehungsberechtigten und Vormünder wieder. Dies geschieht im Kontext der Informationen die der anfragende Nutzer sehen darf.

Die Antwort erfolgt mit dem JSON-Objekt *Listing 3.14*. Die einzelnen Felder der Antwort werden in *Tabelle 3.29* beschrieben.

```

1 | [
2 |   id: "<STRING>",
3 |   ...
4 | ]

```

Listing 3.14: JSON-Antwort für einen GET-Aufruf des Pfads `/api/user/$id/guardians`

Feldname	Datentyp	Beschreibung
	STRING	ID des Erziehungsberechtigten, mit der unter <code>/api/users/\$id</code> weitere Informationen abgefragt werden können.

Tabelle 3.29: Beschreibung der Felder in einem JSON-Objekt für das Zuordnen eines Benutzer in einer Rolle zu einer Schule

## 3.5 Schnittstellen für Unterrichtsfächer an Schulen

Ein Unterrichtsfach ist eine Unterrichtseinheit, welche in einem Schuljahr oder Halbjahr stattfindet und an eine Schule gebunden ist. In der Regel ist es einer Jahrgangsstufe zugeordnet, es kann aber auch mehreren Jahrgangsstufen zugeordnet sein, wenn es sich um eine jahrgangsstufenübergreifende Unterrichtseinheit handelt. Des Weiteren hat ein Unterrichtsfach immer eine Liste an teilnehmenden Schülern und Lehrkräften. Dazu können noch Informationen kommen, welche Klassen an diesen Fach teilnehmen. Weitere optionale Informationen zu einem Unterrichtsfach sind Angaben über den Stundenplan, an welchen Tagen, zu welcher Uhrzeit, wie lange und in welchem Rhythmus das Unterrichtsfach stattfindet.

### 3.5.1 Endpunkt in der REST-API: `/api/subjects`

Die *Tabelle 3.30* listet auf, welche Operationen zugelassen sind und welche HTTP-Methoden dabei verwendet werden.

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

Tabelle 3.30: Zugelassene Operationen auf `/api/subjects`

#### 3.5.1.1 READ

Es sind nur Anfragen mit der HTTP-GET-Methode für ein READ auf die Daten zugelassen. Es werden nur die IDs der Daten präsentiert, die für den anfragenden Benutzer in seinem Kontext existieren.

Die Antwort erfolgt mit dem JSON-Objekt *Listing 3.15*. Die einzelnen Felder der Antwort werden in *Tabelle 3.31* beschrieben.

```

1 [
2   "<STRING>",
3   ...

```



Listing 3.15: JSON-Antwort für einen GET-Aufruf des Pfads `/api/subjects`

Feldname	Datentyp	Beschreibung
	STRING	Zeichenkette, mit der weitere Informationen zu einem Unterrichtsfach unter <code>/api/subjects/\$id</code> abgefragt werden kann.

Tabelle 3.31: Beschreibung der Zeichenkette in der JSON Liste für Unterrichtsfächer.

### 3.5.2 Endpunkt in der REST-API: `/api/subjects/$id`

Die *Tabelle 3.32* listet auf, welche Operationen zugelassen sind und welche HTTP-Methoden dabei verwendet werden.

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

Tabelle 3.32: Zugelassene Operationen auf `/api/subjects/$id`

#### 3.5.2.1 READ

Es sind nur Anfragen mit der HTTP-GET-Methode für ein READ auf die Daten zugelassen. Diese Route gibt bei einer READ-Anfrage für das per ID ausgewählte Unterrichtsfach die allgemeinen Informationen davon wieder. Die Daten, welche wiedergegeben werden, sind durch den Kontext des anfragenden Benutzers eingeschränkt.

Die Antwort erfolgt mit dem JSON-Objekt *Listing 3.16*. Die einzelnen Felder der Antwort werden in *Tabelle 3.33* beschrieben.

```

1 {
2   subject: "<STRING>",
3   name: "<STRING>",
4   school-subject: ["<STRING>", ...],
5   school: "<STRING>",
6   school-year: "<STRING>",
7   start: "<DATE>",
8   end: "<DATE>",
9 }
```

Listing 3.16: JSON-Antwort für einen GET-Aufruf des Pfads `/api/subjects/$id`

Feldname	Datentyp	Beschreibung
subject	STRING	Eindeutige Zeichenkette, die das Unterrichtsfach an einer Schule repräsentiert.
name	STRING	Name des Unterrichtsfaches.
school-subject	Liste von STRING	Für jede Zeichenkette aus dieser Liste kann unter <code>/api/school-subjects/\$id</code> die genauen Informationen abgefragt werden.
school	STRING	ID der Schule, zu der das Fach gehört. Die Daten der Schule können damit unter <code>/api/schools/\$id</code> abgefragt werden.
school-year	STRING	ID des Schuljahres, in dem das Unterrichtsfach angeboten wurde. Das Schuljahr kann damit unter <code>/api/school-years/\$id</code> abgefragt werden.
start	DATE	Datumsangabe, ab dem das Unterrichtsfach angeboten wird.
Ende	DATE	Datumsangabe, bis zu dem das Unterrichtsfach angeboten wird.

Tabelle 3.33: Beschreibung der Felder in einem JSON-Objekt für ein Unterrichtsfach

### 3.5.3 Endpunkt in der REST-API: `/api/subjects/$id/classes`

Die *Tabelle 3.34* listet auf, welche Operationen zugelassen sind und welche HTTP-Methoden dabei verwendet werden.

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

Tabelle 3.34: Zugelassene Operationen auf `/api/subjects/$id/classes`

#### 3.5.3.1 READ

Es sind nur Anfragen mit der HTTP-GET-Methode für ein READ auf die Daten zugelassen. Diese Route gibt bei einer READ-Anfrage eine Liste von Klassen-IDs wieder, welche dem per ID ausgewählten Unterrichtsfach zugeordnet sind. Für die Daten gilt, dass diese anhand des Kontextes des anfragenden Users Einschränkungen unterliegen.

Die Antwort erfolgt mit dem JSON-Objekt *Listing 3.17*. Die einzelnen Felder der Antwort werden in *Tabelle 3.35* beschrieben.

1	[
2	"<STRING>",

```

3 | ...
4 | ]

```

Listing 3.17: JSON-Antwort für einen GET-Aufruf des Pfads `/api/subjects/$id/classes`

Feldname	Datentyp	Beschreibung
	STRING	ID der Klasse, die dem Unterrichtsfach zugeordnet wurde. Die Klasse kann damit unter <code>/api/classes/\$id</code> abgerufen werden.

Tabelle 3.35: Beschreibung der Zeichenkette in der JSON-Liste, welche alle Klassen, die ein Unterrichtsfach haben, umfasst

### 3.5.4 Endpunkt in der REST-API: `/api/subjects/$id/students`

Die *Tabelle 3.36* listet auf, welche Operationen zugelassen sind und welche HTTP-Methoden dabei verwendet werden.

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

Tabelle 3.36: Zugelassene Operationen auf `/api/subjects/$id/students`

#### 3.5.4.1 READ

Es sind nur Anfragen mit der HTTP-GET-Methode für ein READ auf die Daten zugelassen. Eine READ-Anfrage auf diese Route gibt eine Liste mit Objekten, welche Schüler von wann bis wann an dem per ID ausgewählten Unterrichtsfach teilgenommen haben, wieder. Die Daten werden durch den Kontext des anfragenden Benutzers eingeschränkt.

Die Antwort erfolgt mit dem JSON-Objekt *Listing 3.18*. Die einzelnen Felder der Antwort werden in *Tabelle 3.37* beschrieben. Die Berechtigungen auf den Endpoint können *Tabelle ??* entnommen werden.

```

1 | [
2 | {
3 |   subject: "<STRING>",
4 |   user: "<STRING>",
5 |   start: "<DATE>",
6 |   end: "<DATE>",
7 | },
8 | ...
9 | ]

```

Listing 3.18: JSON-Antwort für einen GET-Aufruf des Pfads `/api/subjects/$id/students`

Feldname	Datentyp	Beschreibung
subject	STRING	Zeichenkette, mit der das Unterrichtsfach unter <code>/api/subjects/\$id</code> abgefragt werden kann.
user	STRING	Zeichenkette, mit der User unter <code>/api/users/\$id</code> abgefragt werden kann.
start	DATE	Optional; wird gesetzt, falls der Schüler nach Start des Zeitraumes eines Unterrichtsfaches dem Unterrichtsfach hinzugefügt wurde.
end	DATE	Optional; wird gesetzt, falls ein Schüler vor Ablauf des Zeitraumes eines Unterrichtsfaches aus dem Unterrichtsfach entfernt wurde.

Tabelle 3.37: Beschreibung der Felder in einem JSON-Objekt mit in der Liste aller Schüler eines Unterrichtsfaches.

### 3.5.5 Endpunkt in der REST-API: `/api/subjects/$id/teachers`

Die *Tabelle 3.38* listet auf, welche Operationen zugelassen sind und welche HTTP-Methoden dabei verwendet werden.

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

Tabelle 3.38: Zugelassene Operationen auf `/api/subjects/$id/teachers`

#### 3.5.5.1 READ

Es sind nur Anfragen mit der HTTP-GET-Methode für ein READ auf die Daten zugelassen. Bei einer READ-Abfrage auf diese Route wird eine Liste von Objekten mit Metainformationen zu den unterrichtenden Lehrkräften des per ID ausgewählten Unterrichtsfaches zurückgegeben. Für die Daten gilt, dass diese anhand des Kontextes des anfragenden Users Einschränkungen unterliegen.

Die Antwort erfolgt mit dem JSON-Objekt *Listing 3.19*. Die einzelnen Felder der Antwort werden in *Tabelle 3.39* beschrieben. Die Berechtigungen auf den Endpoint können *Tabelle ??* entnommen werden.

```

2  {
3    subject: "<STRING>",
4    user: "<STRING>",
5    start: "<DATE>",
6    end: "<DATE>",
7  },
8  ...
9 ]

```

Listing 3.19: JSON-Antwort für einen GET-Aufruf des Pfads `/api/subjects/$id/teachers`

Feldname	Datentyp	Beschreibung
subject	STRING	Zeichenkette, mit der das Unterrichtsfach unter <code>/api/subjects/\$id</code> abgefragt werden kann.
user	STRING	Zeichenkette, mit der die Lehrkraft unter <code>/api/users/\$id</code> abgefragt werden kann.
start	DATE	Optional; wird gesetzt, falls die Lehrkraft am Start des Zeitraumes eines Unterrichtsfaches dem Unterrichtsfach hinzugefügt wurde.
end	DATE	Optional; wird gesetzt, falls die Lehrkräfte vor Ablauf des Zeitraumes eines Unterrichtsfaches aus dem Unterrichtsfach entfernt wurde.

Tabelle 3.39: Beschreibung der Felder in einem JSON-Objekt mit der Liste aller unterrichtenden Lehrkräfte eines Unterrichtsfaches.

### 3.5.6 Endpunkt in der REST-API: `/api/subjects/$id/timetable`

Die *Tabelle 3.40* listet auf, welche Operationen zugelassen sind und welche HTTP-Methoden dabei verwendet werden.

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

Tabelle 3.40: Zugelassene Operationen auf `/api/subjects/$id/timetable`

#### 3.5.6.1 READ

Es sind nur Anfragen mit der HTTP-GET-Methode für ein READ auf die Daten zugelassen. Es werden für das per ID ausgewählte Unterrichtsfach die Daten zu für den Stundenplan ausgegeben. Die Daten sind durch den Kontext des anfragenden Users eingeschränkt.

Die Antwort erfolgt mit dem JSON-Objekt *Listing 3.20*. Die einzelnen Felder der Antwort werden in Tabelle *Tabelle 3.41* beschrieben. Die Berechtigungen auf den Endpoint können *Tabelle ??* entnommen werden.

```

1  [
2  {
3    subject: "<STRING>",
4    day: "<ENUM>",
5    start: "<TIME>",
6    end: "<TIME>",
7    repeat: "<ENUM>",
8    date: "<DATE>",
9    week: "<ENUM>",
10  },
11  ...
12 ]

```

Listing 3.20: JSON-Antwort für einen GET-Aufruf des Pfads  
/api/subjects/\$id/timetable

Feldname	Datentyp	Beschreibung
subject	STRING	Zeichenkette, mit der das Unterrichtsfach unter /api/subjects/\$id abgefragt werden kann.
day	ENUM	Tag der Woche, an dem die Unterrichtseinheit für das Unterrichtsfach angeboten wird.
start	TIME	Uhrzeit, ab der die Unterrichtseinheit an dem Tag beginnt.
end	TIME	Uhrzeit, bis zu der die Unterrichtseinheit an dem Tag andauert.
repeat	ENUM	Mögliche Werte sind: <ul style="list-style-type: none"> <li>• “weekly“ für wöchentliche Wiederholung</li> <li>• “biweekly“ für zweiwöchentliche Wiederholung</li> <li>• “onetime“ für einmalige Ansetzung</li> </ul>
date	DATE	Optional; wird gesetzt, falls es sich um den repeat-Typen „onetime“ handelt, und gibt den Tag an, an dem die Unterrichtseinheit stattfindet.

Tabelle 3.41: Beschreibung der Felder in einem JSON-Objekt für den Stundenplan eines Schulfachs

Feldname	Datentyp	Beschreibung
week	ENUM	Optional; wird gesetzt, falls es sich um den repeat-Typen „biweekly“ handelt, <ul style="list-style-type: none"> <li>“week-1“ steht für grade Kalenderwochen</li> <li>“week-2“ steht für ungrade Kalenderwochen</li> </ul>

Tabelle 3.41: Beschreibung der Felder in einem JSON-Objekt für den Stundenplan eines Schulfachs

## 3.6 Schnittstellen für Klassen

### 3.6.1 Endpunkt in der REST-API: /api/classes/\$id

Die *Tabelle 3.42* listet auf, welche Operationen zugelassen sind und welche HTTP-Methoden dabei verwendet werden.

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

Tabelle 3.42: Zugelassene Operationen auf /api/classes/\$id

#### 3.6.1.1 READ

Es sind nur Anfragen mit der HTTP-GET-Methode für ein READ auf die Daten zugelassen. Bei Anfragen an diesen Endpunkt werden die allgemeinen Daten einer Klasse ausgegeben.

Die Antwort erfolgt mit dem JSON-Objekt *Listing 3.21*. Die einzelnen Felder der Antwort werden in *Tabelle 3.43* beschrieben.

```

1 {
2   class: "<STRING>",
3   name: "<STRING>",
4   school: "<STRING>",
5   school-year: "<STRING>",
6   start: "<DATE>",
7   end: "<DATE>",
8   grade: [ "<STRING>", ... ],
9 }
```

Listing 3.21: JSON-Antwort für einen GET-Aufruf des Pfads /api/classes/\$id

Feldname	Datentyp	Beschreibung
class	STRING	ID der Klasse im System.
name	STRING	Name der Klasse, welche zum Anzeigen verwendet wird.
school	STRING	ID, zu welcher Schule die Klasse gehört. Die Schule kann damit unter <code>/api/schools/\$id</code> abgefragt werden.
school-year	STRING	Schuljahr, in welchem die Klasse existiert. Die Daten des Schuljahres können unter <code>/api/school-years/\$id</code> abgerufen werden.
start	DATE	Zeitpunkt, ab wann das Klassenobjekt existiert, falls dies abweichend zum Startzeitpunkt des Schuljahres ist.
end	DATE	Zeitpunkt, bis wann das Klassenobjekt existiert, falls dies abweichend zum Endzeitpunkt des Schuljahres ist.
grade	List of STRINGS	Enthält eine Liste von Klassenstufen von Schülern in dieser Klasse.

Tabelle 3.43: Beschreibung der Felder in einem JSON-Objekt für das Zuordnen eines Benutzer in einer Rolle zu einer Schule

### 3.6.2 Endpunkt in der REST-API: `/api/classes/$id/subjects`

Die *Tabelle 3.44* listet auf, welche Operationen zugelassen sind und welche HTTP-Methoden dabei verwendet werden.

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

Tabelle 3.44: Zugelassene Operationen auf `/api/classes/$id/subjects`

#### 3.6.2.1 READ

Es sind nur Anfragen mit der HTTP-GET-Methode für ein READ auf die Daten zugelassen. Bei Anfragen an diesen Endpunkt wird eine Liste von Fächern, welcher dieser Klasse zugeordnet ist, aufgelistet.

Die Antwort erfolgt mit dem JSON-Objekt *Listing 3.22*. Die einzelnen Felder der Antwort werden in *Tabelle 3.45* beschrieben.

<sup>1</sup> [



```

2  "<STRING>",
3  ...
4  ]

```

Listing 3.22: JSON-Antwort für einen GET-Aufruf des Pfads `/api/classes/$id/subjects`

Feldname	Datentyp	Beschreibung
	STRING	Referenz auf das Unterrichtsfach, kann als ID im Pfad <code>/api/subjects/\$id</code> verwendet werden.

Tabelle 3.45: Beschreibung der Felder in einem JSON-Objekt, welches in der Liste für Unterrichtsfächer einer Klasse enthalten sind.

### 3.6.3 Endpunkt in der REST-API: `/api/classes/$id/students`

Die *Tabelle 3.46* listet auf, welche Operationen zugelassen sind und welche HTTP-Methoden dabei verwendet werden.

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

Tabelle 3.46: Zugelassene Operationen auf `/api/classes/$id/students`

#### 3.6.3.1 READ

Es sind nur Anfragen mit der HTTP-GET-Methode für ein READ auf die Daten zugelassen. Bei Anfragen an diesen Endpunkt wird eine Liste von IDs von Schülern, die in der Klasse sind, zurückgegeben. Optional ist der Eintritts- sowie der Austrittszeitpunkt. Diese Felder werden gesetzt, falls ein Schüler nicht über den kompletten Zeitraum Teil der Klasse war.

Die Antwort erfolgt mit dem JSON-Objekt *Listing 3.23*. Die einzelnen Felder der Antwort werden in *Tabelle 3.47* beschrieben.

```

1  [
2  {
3    user: "<STRING>",
4    start: "<DATE>",
5    end: "<DATE>",
6  },
7  ...
8  ]

```

Listing 3.23: JSON-Antwort für einen GET-Aufruf des Pfads `/api/classes/$id/students`

Feldname	Datentyp	Beschreibung
user	STRING	Die User-ID des Schülers, die Daten des Benutzers können unter <code>/api/users/\$id</code> abgerufen werden
start	DATE	Optional; wird gesetzt, falls ein Schüler später zu eine Klasse hinzugefügt wird.
end	DATE	Optional; wird gesetzt, falls ein Schüler früher aus einer Klasse genommen wird.

Tabelle 3.47: Beschreibung der Felder in einem JSON-Objekt aus der Liste der Schüler einer Klasse

### 3.6.4 Endpunkt in der REST-API: `/api/classes/$id/teachers`

Die *Tabelle 3.48* listet auf, welche Operationen zugelassen sind und welche HTTP-Methoden dabei verwendet werden.

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

Tabelle 3.48: Zugelassene Operationen auf `/api/classes/$id/teachers`

#### 3.6.4.1 READ

Es sind nur Anfragen mit der HTTP-GET-Methode für ein READ auf die Daten zugelassen. Bei Anfragen an diesen Endpunkt wird eine Liste von IDs von Klassenlehrern zurückgegeben, die der Klasse als Klassenlehrer zugeordnet sind. Optional ist der Eintritts- sowie Austrittszeitpunkt. Diese Felder werden gesetzt, falls der Klassenlehrer dies nicht über den kompletten Zeitraum der Klasse war.

Die Antwort erfolgt mit dem JSON-Objekt *Listing 3.24*. Die einzelnen Felder der Antwort werden in *Tabelle 3.49* beschrieben.

```

1  [
2  {
3    user: "<STRING>",
4    start: "<DATE>",
5    end: "<DATE>",
6    order: [
7      {
8        order: "<NUMBER>",
9        start: "<DATE>",
10       end: "<DATE>",
11     },
12     ...

```

```

13 | ],
14 | },
15 | ...
16 | ]

```

Listing 3.24: JSON-Antwort für einen GET-Aufruf des Pfads `/api/classes/$id/teachers`

Feldname	Datentyp	Beschreibung
user	STRING	Die User-ID des Lehrers, die Daten des Benutzers können unter <code>/api/users/\$id</code> abgerufen werden
start	DATE	Optional; wird gesetzt, falls der Lehrer erst während des Zeitraums einer Klasse als Klassenlehrer zugewiesen wird.
end	DATE	Optional; wird gesetzt, falls der Lehrer vor Ende des Zeitraums der Klasse nicht mehr der Klasse als Klassenlehrer zugewiesen ist.
order	Object List	Eine Liste mit Objekten, die angibt, welche Position der Lehrer von wann bis wann in der Klasse innehatte. Niedrigere Nummer heißt höhere Position.
order.order	NUMBER	Die Ordnungsnummer, welche angibt welche Position der Lehrer in einer Klasse inne hat. Beginnt immer bei 1. Es können auch mehrere Lehrer dieselbe Nummer haben, falls die Klasse zum Beispiel mehrere gleichberechtigte Klassenlehrer hat.
order.start	DATE	Optional; Zeitpunkt, ab dem der Lehrer die Position innehatte.
order.end	DATE	Optional; Zeitpunkt, bis zu dem der Lehrer die Position innehatte.

Tabelle 3.49: Beschreibung der Felder in einem JSON-Objekt aus der Liste der Lehrer

### 3.6.5 Endpunkt in der REST-API: `/api/classes/$id/representatives`

Die *Tabelle 3.50* listet auf, welche Operationen zugelassen sind und welche HTTP-Methoden dabei verwendet werden.

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

Tabelle 3.50: Zugelassene Operationen auf `/api/classes/$id/representatives`

### 3.6.5.1 READ

Es sind nur Anfragen mit der HTTP-GET-Methode für ein READ auf die Daten zugelassen. Bei Anfragen an diesen Endpunkt wird eine Liste von Schülern oder Vormündern, welche in einer Klasse oder Jahrgangsstufe die Statusgruppe repräsentieren, zurückgegeben. Die Position wird durch eine Ordnungsnummer widerspiegelt. Je kleiner die Nummer, desto wichtiger der Repräsentant der Statusgruppe. Es können auch mehrere Repräsentanten einer Statusgruppe die gleiche Ordnungsnummer haben. Zusätzlich kann Eintrittszeitpunkt sowie der Austrittszeitpunkt vermerkt werden, falls dieser abweichend zum Zeitraum der Klasse ist.

Die Antwort erfolgt mit dem JSON-Objekt *Listing 3.25*. Die einzelnen Felder der Antwort werden in *Tabelle 3.51* beschrieben.

```
1 [
2   {
3     user: "<STRING>",
4     role: "<STRING>",
5     order: "<NUMBER>",
6     start: "<DATE>",
7     end: "<DATE>",
8   },
9   ...
10 ]
```

Listing 3.25: JSON-Antwort für einen GET-Aufruf des Pfads  
/api/classes/\$id/representatives

Feldname	Datentyp	Beschreibung
user	STRING	Die User-ID des Repräsentanten der Statusgruppe, die Daten des Benutzers können unter /api/users/\$id abgerufen werden
role	STRING	Die Rolle, welcher der Repräsentant der Statusgruppe im System hat (zugelassen sind student oder guardian).
order	NUMBER	Die Ordnungsnummer, die angibt, welche Position der Repräsentant der Statusgruppe in einer Klasse innehat.
start	DATE	Optional; wird benötigt, wenn der Repräsentant der Statusgruppe zu einem abweichenden Zeitpunkt das Amt bekommt.
end	DATE	Optional; wird benötigt, wenn der Repräsentant der Statusgruppe vorzeitig vor Ende des Zeitraums der Klasse nicht mehr als Repräsentant der Statusgruppe fungiert.

Tabelle 3.51: Beschreibung der Felder in einem JSON-Objekt aus der Liste der Repräsentanten von Statusgruppen in einer Klasse

## 4 Weiterführende Konzepte

### 4.1 Löschkonzept

Bei Verwendung der in diesem Dokument definierten REST-API muss der Client ein Löschkonzept auf die über die REST-API bezogenen Daten definieren bzw. diesem mit dem IDM-Provider abstimmen.

Der Client hat die Möglichkeit, über ein Benutzerkonto mit der Rolle “sync-system“ die im Client vorhandenen Datenobjekte und Referenzen darauf über die REST-API beim IDM-Provider zu revalidieren. Existiert diese Möglichkeit der Rücksynchronisation im Client nicht, so müssen Überlegungen getroffen werden, wann Daten als veraltet gelten und nach welchen Aufbewahrungsfristen diese auf Client-Seite gelöscht werden. Für Benutzerkonten kann ein Kriterium sein, wie lange sich eine Person nicht mehr im Client eingeloggt hat, bevor MaSSnahmen getroffen werden, den Datenstand aktuell zu halten (Ankündigung der Deaktivierung/Löschung des Benutzerkontos, Löschung des Benutzerkontos).

Bei Löschung eines Benutzerkontos im Client ist zu prüfen, inwiefern Daten vorliegen, die dem geistigen Eigentum der Inhaberin bzw. des Inhabers des Benutzerkontos unterliegen, und diese für einen definierten Zeitraum nach Löschung des Benutzerkontos durch die Inhaberin bzw. den Inhaber zur Sicherung abrufbar sind.

### 4.2 Glossar

#### 4.2.1 Begriffe der Fachdomäne

##### **Schüler**

Unterrichtsteilnehmer an einer Schule

##### **Externe Schüler**

Schüler, die einer anderen Schule angehören und bei einer anderen Schule schulübergreifende Unterrichtsangebote wahrnehmen.

##### **Eltern**

die für die Person der minderjährigen Schülerin oder des minderjährigen Schülers einzeln oder gemeinsam Sorgeberechtigten oder ihnen nach diesem Gesetz gleichgestellte

Personen

### **Lehrer**

Lehrer und pädagogische Fachkräfte an Schulen, die Unterricht erteilen

### **Schulleitung**

Lehrkraft einer Schule sowie ggf. weitere Vertreter aus dem Lehrerkollegium

### **Schulen**

Einrichtungen, in denen unabhängig vom Wechsel der Lehrkräfte, Schülerinnen und Schüler durch planmäßiges und gemeinsames Lernen und durch das gemeinsame Schulleben bestimmte Erziehungs- und Bildungsziele erreicht werden sollen.

### **Schulträger**

Eine juristische oder natürliche Person, die für die Errichtung, Organisation und Verwaltung der einzelnen Schule rechtlich unmittelbar die Verantwortung trägt und zur Unterhaltung der Schule eigene Leistungen erbringt.

### **Schuljahr**

Beschreibt einen Zeitraum in dem Klassenstufen und Klassen für gewöhnlich definiert sind.

### **Klasse**

Lerngemeinschaft aus Schülern, welche für einen Zeitraum (z.B. Schuljahr 2020/21) an einer Schule unter Betreuung von Lehrkräften am Unterricht im Klassenverband teilnimmt. Eine Klassenstufe umfasst mehrere Klassen einer Stufe, z.B. (5a, 5b, 5c im Schuljahr 2020/21).

### **Unterrichtsfach**

In unterschiedlichen Fächern werden Unterrichtsinhalte den Schülern durch eine oder mehrere Lehrkräfte zu bestimmten Zeiten in einem bestimmten Zeitraum vermittelt.

## **4.2.2 Technische Begriffe**

### **REST API**

Technische Schnittstelle zum Austausch von Daten über HTTP-Methoden.

**Endpunkt**

URL-Pfad innerhalb einer REST API, unter dem bestimmte Daten abgerufen können.

**Authentifizierung**

Zur Nachvollziehbarkeit des Zugriffs und generischem Zugriffsschutz sind Endpunkte durch technische Konzepte geschützt, die den öffentlichen Zugriff beschränken und bei API-Zugriff die Zuordnung eines (technischen) Nutzers ermöglichen.

**Authorisierung**

Zum Schutz der über einen Endpunkt abrufbaren Daten kann die Berechtigung auf und Sichtbarkeit von Datensätzen je nach Rolle eines authentifiziertem Clients eingeschränkt werden.

**IDM****Client**

Eine Anwendung, welche über die REST API lesend auf Daten des IDM zugreift oder diese verändert.

**JavaScript Object Notation (JSON))**

Proprietäres Datenformat zum Austausch von Daten über teilweise menschenlesbare Textform.

# Abbildungsverzeichnis

2.1	Ablauf des Authentication Code Grant . . . . .	14
-----	--	----



# Tabellenverzeichnis

1.1	Mögliche Benutzerrollen im Kontext SSSchule" . . . . .	12
2.1	Endpunkte, die durch den IDM-Provider für die Anmeldung per OAuth2 zur Verfügung gestellt werden müssen . . . . .	13
3.1	Liste der abgestimmten Referenzschulfächer . . . . .	16
3.2	Zugelassene Operationen auf /api/school-subjects . . . . .	16
3.3	Beschreibung der Felder in einem JSON-Objekt für ein Schulfach . . . . .	17
3.4	Zugelassene Operationen auf /api/school-years . . . . .	17
3.5	Beschreibung der Felder in einem JSON-Objekt aus der Liste der Schuljahre . . . . .	18
3.6	Zugelassene Operationen auf /api/schools . . . . .	18
3.7	Beschreibung der Felder in einem JSON-Objekt, was die Schul-IDs wiedergibt . . . . .	19
3.8	Zugelassene Operationen auf /api/schools/\$id . . . . .	19
3.9	Beschreibung der Felder in einem JSON-Objekt für das Zuordnen eines Benutzer in einer Rolle zu einer Schule . . . . .	20
3.10	Zugelassene Operationen auf /api/schools/\$id/users . . . . .	20
3.11	Beschreibung der Felder in einem JSON-Objekt für das Zuordnen eines Benutzer in einer Rolle zu einer Schule . . . . .	21
3.12	Zugelassene Operationen auf /api/schools/\$id/classes . . . . .	21
3.13	Beschreibung der Felder in einem JSON-Objekt, welches die Klassen an einer Schule repräsentiert. . . . .	22
3.14	Zugelassene Operationen auf /api/schools/\$id/subjects . . . . .	22
3.15	Beschreibung der Felder in einem JSON-Objekt, welches die Unterrichtsfächer an einer Schule repräsentieren . . . . .	23
3.16	Zugelassene Operationen auf /api/users . . . . .	23
3.17	Beschreibung der Felder in einem JSON-Objekt von einem Benutzer . . . . .	24
3.18	Zugelassene Operationen auf /api/users/\$id . . . . .	24
3.19	Beschreibung der Felder in einem JSON-Objekt von einem Benutzer . . . . .	25
3.20	Zugelassene Operationen auf /api/users/\$id/memberships . . . . .	25
3.21	Beschreibung der Felder in einem JSON-Objekt für das Zuordnen eines Benutzer in einer Rolle zu einer Schule . . . . .	26
3.22	Zugelassene Operationen auf /api/users/\$id/classes . . . . .	26
3.23	Beschreibung der Felder in einem JSON-Objekt aus der Liste der Klassen eines Benutzers . . . . .	27
3.24	Zugelassene Operationen auf /api/users/\$id/subjects . . . . .	28

3.25	Beschreibung der Zeichenkette in der JSON-Liste für Unterrichtsfächer eines Benutzers . . . . .	28
3.26	Zugelassene Operationen auf /api/users/\$id/children . . . . .	28
3.27	Beschreibung der Zeichenkette in der JSON-Liste für Kinder von Erziehungsberechtigten. . . . .	29
3.28	Zugelassene Operationen auf /api/users/\$id/guardians . . . . .	29
3.29	Beschreibung der Felder in einem JSON-Objekt für das Zuordnen eines Benutzer in einer Rolle zu einer Schule . . . . .	30
3.30	Zugelassene Operationen auf /api/subjects . . . . .	30
3.31	Beschreibung der Zeichenkette in der JSON Liste für Unterrichtsfächer. . .	31
3.32	Zugelassene Operationen auf /api/subjects/\$id . . . . .	31
3.33	Beschreibung der Felder in einem JSON-Objekt für ein Unterrichtsfach . .	32
3.34	Zugelassene Operationen auf /api/subjects/\$id/classes . . . . .	32
3.35	Beschreibung der Zeichenkette in der JSON-Liste, welche alle Klassen, die ein Unterrichtsfach haben, umfasst . . . . .	33
3.36	Zugelassene Operationen auf /api/subjects/\$id/students . . . . .	33
3.37	Beschreibung der Felder in einem JSON-Objekt mit in der Liste aller Schüler eines Unterrichtsfaches. . . . .	34
3.38	Zugelassene Operationen auf /api/subjects/\$id/teachers . . . . .	34
3.39	Beschreibung der Felder in einem JSON-Objekt mit der Liste aller unterrichtenden Lehrkräfte eines Unterrichtsfaches. . . . .	35
3.40	Zugelassene Operationen auf /api/subjects/\$id/timetable . . . . .	35
3.41	Beschreibung der Felder in einem JSON-Objekt für den Stundenplan eines Schulfachs . . . . .	36
3.41	Beschreibung der Felder in einem JSON-Objekt für den Stundenplan eines Schulfachs . . . . .	37
3.42	Zugelassene Operationen auf /api/classes/\$id . . . . .	37
3.43	Beschreibung der Felder in einem JSON-Objekt für das Zuordnen eines Benutzer in einer Rolle zu einer Schule . . . . .	38
3.44	Zugelassene Operationen auf /api/classes/\$id/subjects . . . . .	38
3.45	Beschreibung der Felder in einem JSON-Objekt, welches in der Liste für Unterrichtsfächer einer Klasse enthalten sind. . . . .	39
3.46	Zugelassene Operationen auf /api/classes/\$id/students . . . . .	39
3.47	Beschreibung der Felder in einem JSON-Objekt aus der Liste der Schüler einer Klasse . . . . .	40
3.48	Zugelassene Operationen auf /api/classes/\$id/teachers . . . . .	40
3.49	Beschreibung der Felder in einem JSON-Objekt aus der Liste der Lehrer .	41
3.50	Zugelassene Operationen auf /api/classes/\$id/representatives . . . . .	42
3.51	Beschreibung der Felder in einem JSON-Objekt aus der Liste der Repräsentanten von Statusgruppen in einer Klasse . . . . .	42
3.51	Beschreibung der Felder in einem JSON-Objekt aus der Liste der Repräsentanten von Statusgruppen in einer Klasse . . . . .	43

# Listings

1	Beispiel Benutzer mit Rolle 'student' . . . . .	1
2	Beispiel für Benutzer mit Rollen 'teacher' und 'guardian' . . . . .	3
3	Beispiel eines Schulfachs . . . . .	4
4	Klassen-Datenmodell Beispiel 2: Grundschulklasse . . . . .	5
5	Klassen-Datenmodell Beispiel 2: Jahrgangstufe 11, Sekundarstufe 2 . . . .	7
6	Klassen-Datenmodell Beispiel 3: Jahrgangsübergreifende Klasse . . . . .	8
7	Schulen-Datenmodell Beispiel 1 . . . . .	9
2.1	Beispielhafter Aufruf des Authorization Endpoints . . . . .	15
3.1	JSON-Antwort für einen GET-Aufruf des Pfads /api/school-subjects . . .	17
3.2	JSON-Antwort für einen GET-Aufruf des Pfads /api/school-years . . . .	18
3.3	JSON-Antwort für einen GET-Aufruf des Pfads /api/schools . . . . .	19
3.4	JSON-Antwort für einen GET-Aufruf des Pfads /api/school/\$id . . . . .	19
3.5	JSON-Antwort für einen GET-Aufruf des Pfads /api/schools/\$id/users . .	20
3.6	JSON-Antwort für einen GET-Aufruf des Pfads /api/schools/\$id/classes .	22
3.7	JSON-Antwort für einen GET-Aufruf des Pfads /api/schools/\$id/subjects	22
3.8	JSON-Antwort für einen GET-Aufruf des Pfads /api/users . . . . .	23
3.9	JSON-Antwort für einen GET-Aufruf des Pfads /api/users/\$id . . . . .	24
3.10	JSON-Antwort für einen GET-Aufruf des Pfads /api/users/\$id/assignments	26
3.11	JSON-Antwort für einen GET-Aufruf des Pfads /api/users/\$id/classes . .	27
3.12	JSON-Antwort für einen GET-Aufruf des Pfads /api/users/\$id/subjects .	28
3.13	JSON-Antwort für einen GET-Aufruf des Pfads /api/users/\$id/children .	29
3.14	JSON-Antwort für einen GET-Aufruf des Pfads /api/user/\$id/guardians .	29
3.15	JSON-Antwort für einen GET-Aufruf des Pfads /api/subjects . . . . .	30
3.16	JSON-Antwort für einen GET-Aufruf des Pfads /api/subjects/\$id . . . .	31
3.17	JSON-Antwort für einen GET-Aufruf des Pfads /api/subjects/\$id/classes	32
3.18	JSON-Antwort für einen GET-Aufruf des Pfads /api/subjects/\$id/students	33
3.19	JSON-Antwort für einen GET-Aufruf des Pfads /api/subjects/\$id/teachers	34
3.20	JSON-Antwort für einen GET-Aufruf des Pfads /api/subjects/\$id/time- table . . . . .	36
3.21	JSON-Antwort für einen GET-Aufruf des Pfads /api/classes/\$id . . . . .	37
3.22	JSON-Antwort für einen GET-Aufruf des Pfads /api/classes/\$id/subjects	39
3.23	JSON-Antwort für einen GET-Aufruf des Pfads /api/classes/\$id/students	39
3.24	JSON-Antwort für einen GET-Aufruf des Pfads /api/classes/\$id/teachers	40
3.25	JSON-Antwort für einen GET-Aufruf des Pfads /api/classes/\$id/repre- sentatives . . . . .	42