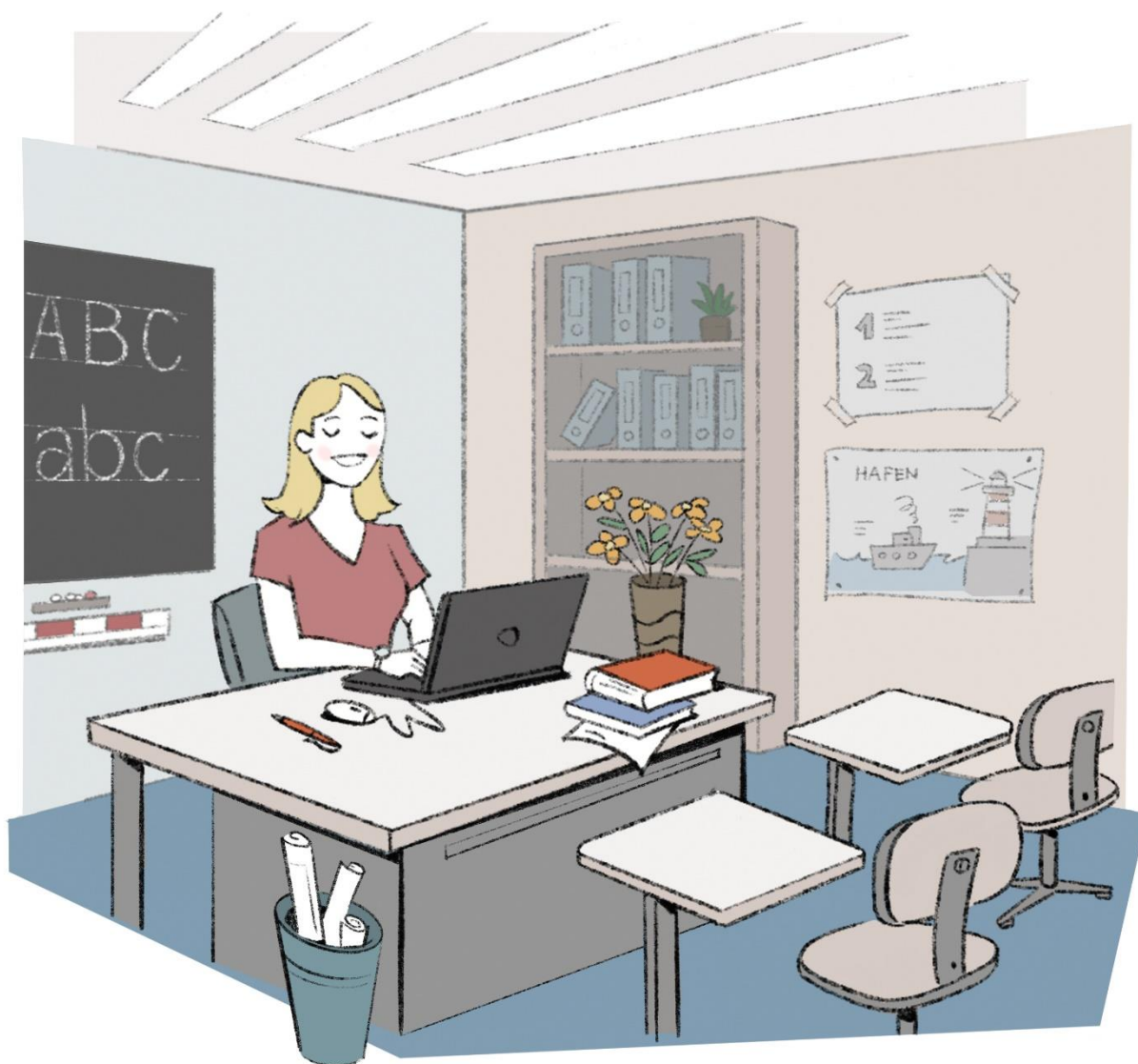


Stand: 24.11.2020

Datenschnittstelle XML Definition



Über dieses Dokument

Versionierung

Version	Datum	Autor	Status
0.1	25.03.2011	Stefan Roth	Dokument erstellt
0.2	30.09. 2011	Stefan Roth	Daten einer Arbeitsgruppe (Fachklasse/Niveaunklasse)
0.3	19.12. 2011	Stefan Roth	Versionsverzeichnis in diesem Dokument eingefügt
0.4	12.01. 2012	Stefan Roth	Neue Elemente 'passover' für Angaben zum Übertritt (Seite 89)
0.5	27.01. 2012	Tino Gruse	Neues Attribut 'id' als Fremdschlüssel für Lernende und Lehrpersonen
0.6	28.03. 2012	Tino Gruse	SOAP-Sektion korrigiert
0.7	11.07.2012	Tino Gruse	Attribut „bur“ von Element „inst“ entfernt. Attribut „customerKey“ im Element „inst“ als zwingend notwendige Kundennummer hinzugefügt (Seite 30) Element „bur“ in Element „activity“ hinzugefügt (Seite 47) Prozesse erweitert mit Definitionen der Schulklassen und Gruppen (Seite 12) Sektion „Definitionen aller Schulklassen“ hinzugefügt (Seite 53) Attribut „complete“ in Element „schoolclasses“ hinzugefügt (Seite 56) Element „bur“ in Element „classDefinition“ hinzugefügt (Seite 54) Element „schoolhouseKey“ in Element „classData“ hinzugefügt (Seite 57) Sektion „Definitionen aller Gruppen“ hinzugefügt (Seite 110) Attribut „complete“ in Element „studyGroups“ hinzugefügt (Seite 117) Element „unitName“ in Element „groupDefinition“ hinzugefügt (Seite 120) Element „bur“ in Element „groupDefinition“ hinzugefügt (Seite 120) Element „schoolhouseKey“ in Element „groupData“ hinzugefügt (Seite122)
0.8	13.08.2012	Tino Gruse	Sektion „SOAP-Fehler“ hinzugefügt (Seite 16) Element „classFormCode“ in Element „classDefinition“ hinzugefügt Element „classFormName“ in Element „classDefinition“ hinzugefügt Element „room“ in Element „classData“ hinzugefügt Element „room“ in Element „groupData“ hinzugefügt
0.9	17.12.2012	Tino Gruse	Element „group“ in Element „subject“ hinzugefügt (Seite 108)
0.10	18.12.2012	Tino Gruse	Element „schoolFormCode“ in Element „schoolData“ hinzugefügt (Seite 101) Element „schoolFormName“ in Element „schoolData“ hinzugefügt (Seite 101)
0.11	10.01.2014	Tino Gruse	Element „schoolClass“ in Element „entrance“ hinzugefügt (Seite 83) Element „reason“ in Element „leaving“

			hinzugefügt (Seite 86) Import von Förderlehrpersonen hinzugefügt (Seite 71)
0.12	13.03.2014	Tino Gruse	Attribut 'target' im Element 'lehreroffice' sowie Element 'mainFunction' unter 'work' hinzugefügt
0.13	07.04.2014	Tino Gruse	Elemente 'DAZ' und 'VM' im Element 'student/school-Data' sowie Element 'ILZ' unter 'student/subjects/subject' hinzugefügt Element 'unitCity' im Element 'classDefinition' sowie 'groupDefinition' und Element 'schoolhouseCity' unter 'classData' sowie 'groupData' hinzugefügt
0.14	13.07.2015	Tino Gruse	Element 'isEmergencyContact' im Element 'student/contacts/contact' hinzugefügt Attribut 'caption' im Element 'freeField' hinzugefügt
0.15	24.06.2016	Tino Gruse Stefan Roth	Neuen Abschnitt 'individualFostering' mit diversen Fördermassnahmen hinzugefügt
0.16	26.10.2016	Simone Sterren	Element 'emailSchool' im Element 'teacher/work/workadress' hinzugefügt Korrektur bei der Anzahl Lehrpersonen für den Import (Seite 75)
0.17	19.05.2017	Tino Gruse Stefan Roth	Element 'mediaRepresentation' sowie 'transport' und 'distance' im Element 'student/homeAddress' hinzugefügt
0.18	11.12.2017	Tino Gruse Simone Sterren	Element 'personalNumber' im Element 'teacher/work' hinzugefügt
0.19	18.01.2019	Tino Gruse Simone Sterren	Attribut für freiwilliges Kindergartenjahr hinzugefügt sowie Attribut 'caption' bei Schulfächern hinzugefügt
0.20	02.08.2019	Tino Gruse	Element 'municipality' im Element 'student-Data/homeAddress' hinzugefügt
0.21	26.08.2019	Simone Sterren	Informationen, was von Seiten CMI LehrerOffice implementiert ist, nachgeführt
0.22	10.01.2020	Simone Sterren	Element 'individualFostering' mit Elementen für Ereignisse (Abklärungen oder Massnahmen) erweitert und einen Abschnitt für den Import von Absenzen eingefügt
0.23	30.01.2020	Simone Sterren	Attribut 'ilzComplete' im Element 'student/subjects' hinzugefügt.
0.24	04.06.2020	Jürg Otter	Elemente unter 'individualFostering' für Kanton GR ergänzt: anstelle <if> ist im Kanton GR <ifol> und <ifml> zu verwenden, damit ein Mapping für die Schullaufbahn möglich ist.
0.25	24.11.2020	Simone Sterren	Element 'profession' aktualisiert

Inhaltsverzeichnis

Über dieses Dokument	2
Inhaltsverzeichnis	4
1 Grundlegendes	10
1.1 Einleitung	10
1.2 Grobkonzept.....	10
1.3 Prozesse	11
1.3.1 Daten für Lehrpersonen.....	11
1.3.2 Definitionen aller Schulklassen.....	11
1.3.3 Daten einer Schulklasse	11
1.3.4 Definitionen aller Gruppen / Fachklassen.....	12
1.3.5 Daten einer Arbeitsgruppe / Fachklasse.....	12
1.3.6 Weitere Prozesse	12
1.4 SOAP – Anforderung	13
1.4.1 Element 'soap:Envelope'	14
1.4.2 Namespaces zum Element 'soap:Envelope'.....	14
1.4.3 Element 'soap:Header'.....	15
1.4.4 Element 'ServiceAuthHeader'	16
1.4.5 Attribute zum Element 'ServiceAuthHeader'.....	16
1.4.6 Elemente zu 'ServiceAuthHeader'	17
1.4.7 Element 'soap:Body'	18
1.5 SOAP – Antwort.....	19
1.5.1 Element 'soap:Envelope'	20
1.5.2 Namespaces zum Element 'soap:Envelope'.....	20
1.5.3 Element 'soap:Body'	21
1.6 SOAP – Fehler.....	22
1.6.1 Element 'LO_error'.....	22
1.6.2 Elemente zu 'LO_Error'.....	22
2 Daten der Lehrpersonen	24
2.1 SOAP – Anfrage für Daten der Lehrpersonen	25
2.1.1 Element 'LO_getTeachers'	25
2.1.2 Attribute zum Element 'LO_getTeachers'	25
2.1.3 Elemente zu ' LO_getTeachers ' als Parameter	25
2.2 Daten für Lehrpersonen – XML-Struktur	27

2.3	Daten für Lehrpersonen – Elemente	28
2.3.1	Element 'lehreroffice'	28
2.3.2	Namespaces zum Element 'lehreroffice'	28
2.3.3	Attribute zum Element 'lehreroffice'	28
2.3.4	Element 'inst'	30
2.3.5	Attribute zum Element 'inst'	30
2.3.6	Element 'source'	32
2.3.7	Elemente zu 'source'	32
2.3.8	Element 'semesters'	33
2.3.9	Element 'semester'	33
2.3.10	Attribute zum Element 'semester'	33
2.3.11	Element 'teachers'	34
2.3.12	Attribute zum Element 'teachers'	34
2.3.13	Element 'teacher'	35
2.3.14	Attribute zum Element 'teacher'	35
2.3.15	Element 'personalData'	37
2.3.16	Elemente zu 'personalData'	37
2.3.17	Element 'homeAddress'	39
2.3.18	Elemente zu 'homeAddress'	39
2.3.19	Element 'freeFields'	41
2.3.20	Elemente zu 'freeFields'	41
2.3.21	Element 'work'	42
2.3.22	Elemente zu 'work'	42
2.3.23	Element 'workAddress'	43
2.3.24	Elemente zu 'workAddress'	43
2.3.25	Element 'activities'	46
2.3.26	Attribute zum Element 'activities'	46
2.3.27	Element 'activity'	47
2.3.28	Attribute zum Element 'activity'	47
2.3.29	Elemente zu 'activity'	47
2.3.30	Element 'login'	50
2.3.31	Attribute zum Element 'login'	50
2.3.32	Elemente zu 'login'	51
3	Definitionen aller Schulklassen.....	52
3.1	SOAP – Anfrage für die Definitionen aller Schulklassen.....	53
3.1.1	Element 'LO_getSchoolclassDefinitions'	53
3.1.2	Attribute zum Element 'LO_getSchoolclassDefinitions'	53

3.1.3	Elemente zu ' LO_getSchoolclassDefinitions ' als Parameter	54
3.2	Definitionen aller Schulklassen – XML-Struktur	55
3.3	Definitionen aller Schulklassen – Elemente	55
3.3.1	Element 'schoolclasses'	56
3.3.2	Attribute zum Element 'schoolclasses'	56
3.3.3	Element 'schoolclass'	57
3.3.4	Attribute zum Element 'schoolclass'	57
3.3.5	Element 'classDefinition'	59
3.3.6	Attribute zu 'classDefinition'	59
3.3.7	Elemente zu 'classDefinition'	59
3.3.8	Element 'classData'	62
3.3.9	Elemente zu 'classData'	62
4	Daten einer Schulklasse mit Schülerdaten	63
4.1	SOAP – Anfrage für Daten einer Schulklasse mit Schülerdaten	64
4.1.1	Element 'LO_getSchoolclass'	64
4.1.2	Attribute zum Element 'LO_getSchoolclass'	64
4.1.3	Elemente zu ' LO_getSchoolclass ' als Parameter	65
4.2	Daten einer Schulklasse – XML-Struktur	66
4.3	Daten einer Schulklasse – Elemente	68
4.3.1	Element 'schoolclasses'	68
4.3.2	Element 'schoolclass'	69
4.3.3	Attribute zum Element 'schoolclass'	69
4.3.4	Element 'classDefinition'	71
4.3.5	Attribute zu 'classDefinition'	71
4.3.6	Elemente zu 'classDefinition'	71
4.3.7	Element 'classData'	74
4.3.8	Elemente zu 'classData'	74
4.3.9	Element 'classTeachers'	75
4.3.10	Attribute zum Element 'classTeachers'	75
4.3.11	Element 'classTeacher'	76
4.3.12	Attribute zum Element 'classTeacher'	76
4.3.13	Element 'subjects'	77
4.3.14	Element 'subject'	77
4.3.15	Attribute zum Element 'subject'	77
4.3.16	Element 'subjectTeachers'	78
4.3.17	Attribute zum Element 'subjectTeachers'	78
4.3.18	Element 'subjectTeacher'	79

4.3.19 Attribute zum Element 'subjectTeacher'	79
4.3.20 Element 'students'	80
4.3.21 Attribute zum Element 'students'	80
4.3.22 Element 'student'	81
4.3.23 Attribute zum Element 'student'	81
4.3.24 Element 'studentData'	82
4.3.25 Elemente zu 'studentData'	82
4.3.26 Element 'homeAddress'	85
4.3.27 Elemente zu 'homeAddress'	85
4.3.28 Element 'freeFields'	87
4.3.29 Elemente zu 'freeFields'	87
4.3.30 Elemente zum Element 'entrance'	88
4.3.31 Attribute zum Element 'passover'	89
4.3.32 Elemente zum Element 'passover'	89
4.3.33 Elemente zum Element 'leaving'	91
4.3.34 Attribute zum Element 'contacts'	92
4.3.35 Element 'contact'	93
4.3.36 Attribute zum Element 'contact'	93
4.3.37 Elemente zu 'contact'	93
4.3.38 Element 'homeAddress'	96
4.3.39 Elemente zu 'homeAddress'	96
4.3.40 Element 'workAddress'	98
4.3.41 Elemente zu 'workAddress'	98
4.3.42 Element 'history'	101
4.3.43 Attribute zum Element 'history'	101
4.3.44 Element 'addressitem'	102
4.3.45 Elemente zu 'addressitem'	102
4.3.46 Element 'schoolData'	104
4.3.47 Elemente zu 'schoolData'	104
4.3.48 Element 'individualFostering'	107
4.3.49 Elemente zu 'individualFostering'	107
4.3.50 Element 'occurrences'	109
4.3.51 Attribute zum Element 'occurrences'	109
4.3.52 Attribute zum Element 'occurrence'	109
4.3.53 Elemente zum Element 'occurrence'	109
4.3.54 Element 'subjects'	111
4.3.55 Attribute zum Element 'subjects'	111
4.3.56 Element 'subject'	112

4.3.57	Attribute zum Element 'subject'	112
4.3.58	Elemente zu 'subject'	112
4.3.59	Element 'absences'	114
4.3.60	Attribute zum Element 'absences'	114
4.3.61	Element 'absence'	114
4.3.62	Attribute zum Element 'absence'	114
4.3.63	Elemente zu 'absence'	115
4.3.64	Element 'excused'	116
4.3.65	Elemente zu 'excused'	116
5	Definitionen aller Gruppen (Fachklassen / Niveaunklassen)	117
5.1	SOAP – Anfrage für Definitionen aller Gruppen	118
5.1.1	Element 'LO_getStudyGroupDefinitions'	118
5.1.2	Attribute zum Element 'LO_getStudyGroupDefinitions'	118
5.1.3	Elemente zu 'LO_getStudyGroupDefinitions' als Parameter	119
5.2	Definitionen aller Gruppen – XML-Struktur	120
5.3	Definitionen aller Gruppen – Elemente	120
5.3.1	Element 'studyGroups'	121
5.3.2	Attribute zum Element 'studyGroups'	121
5.3.3	Element 'studyGroup'	122
5.3.4	Attribute zum Element 'studyGroup'	122
5.3.5	Element 'groupDefinition'	124
5.3.6	Attribute zu 'groupDefinition'	124
5.3.7	Elemente zu 'groupDefinition'	124
5.3.8	Element 'groupData'	126
5.3.9	Elemente zu 'groupData'	126
6	Daten einer Arbeitsgruppe (Fachklasse / Niveaunklasse)	127
6.1	SOAP – Anfrage für Daten einer Arbeitsgruppe	128
6.1.1	Element 'LO_getStudyGroup'	128
6.1.2	Attribute zum Element 'LO_getStudyGroup'	128
6.1.3	Elemente zu 'LO_getStudyGroup' als Parameter	129
6.2	Daten einer Arbeitsgruppe – XML-Struktur	130
6.3	Daten einer Arbeitsgruppe – Elemente	131
6.3.1	Element 'studyGroups'	132
6.3.2	Element 'studyGroup'	133
6.3.3	Attribute zum Element 'studyGroup'	133
6.3.4	Element 'groupDefinition'	135

6.3.5	Attribute zu 'groupDefinition'	135
6.3.6	Elemente zu 'groupDefinition'	135
6.3.7	Element 'groupData'	137
6.3.8	Elemente zu 'groupData'	137
6.3.9	Element 'groupTeachers'	138
6.3.10	Attribute zum Element 'groupTeachers'	138
6.3.11	Element 'groupTeacher'	139
6.3.12	Attribute zum Element 'groupTeacher'	139
6.3.13	Element 'students'	140
6.3.14	Attribute zum Element 'students'	140
6.3.15	Element 'student'	141
6.3.16	Attribute zum Element 'student'	141
7	Anhang	142
7.1	XML-Entities	142
7.2	Rechtliche Hinweise und Urheberrechte	143

Für die automatische Datenübernahme von Schulverwaltungslösungen stellt CMI LehrerOffice eine Dienst-anwendung zur Verfügung, womit Daten im vorgegebenen XML-Format übernommen werden können. Das zwingend notwendige XML-Format ist in diesem Dokument beschrieben.

1 Grundlegendes

1.1 Einleitung

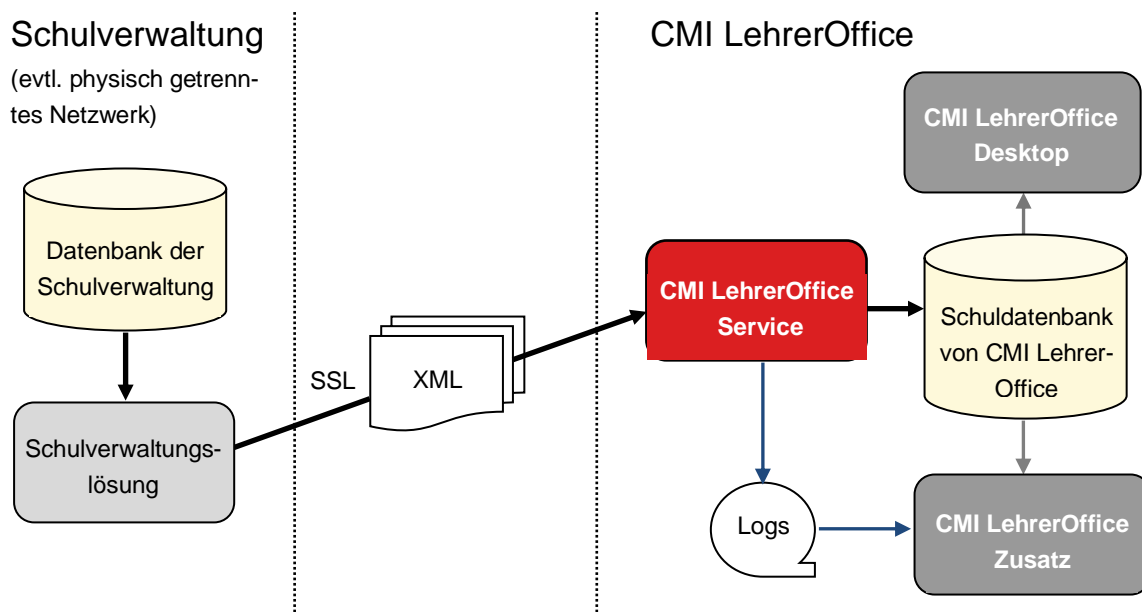
CMI LehrerOffice bietet zwei Arten der Datenübertragung an:

- Dateitransfer in einen bestimmten Zielordner (z.B. FTPS)
- Datenabruf über Webservice (HTTPS SOAP 1.2)

Das Datenformat ist beiden Varianten identisch und ist eine ausbaubare XML-Datei. Der Aufbau dieser XML-Datei ist von CMI LehrerOffice definiert und gilt für alle Schulverwaltungslösungen. Bei der Definition der XML-Datei wurden die Empfehlungen eCH (<http://www.ech.ch>) berücksichtigt:

http://www.ech.ch/index.php?option=com_docman&task=doc_download&gid=332&Itemid=226&lang=de

1.2 Grobkonzept



Wenn der Dateitransfer nicht mittels SOAP stattfinden soll, müssen die Exportdateien als XML-Dateien in einen bestimmten Ordner abgelegt werden, von welchem die Dateien von CMI LehrerOffice zum gegebenen Zeitpunkt automatisch eingelesen werden. Dabei macht es Sinn, jeweils die Daten vom aktuellen sowie vom künftigen Semester zu exportieren, so dass die Vorbereitungsarbeiten für das nachfolgende Halbjahr vorzeitig begonnen werden können.

Beim Export mittels Dateien ist der Syntax der Dateinamen zu beobachten, damit die korrekte Verarbeitungsreihenfolge garantiert werden kann. Die Dateinamen sind im nachfolgenden Abschnitt definiert.

1.3 Prozesse

Zurzeit werden folgende zwei Prozesse resp. Datenübernahmen von der Schulverwaltung unterstützt, wobei die Elemente und dessen Attribute nachfolgend detailliert umschrieben werden.

Obwohl die XML-Struktur Daten von mehreren Semestern und Klassen zulässt, empfehlen wir pro Klasse und Semester eine eigene, abgeschlossene XML-Datei zu verwenden.

1.3.1 Daten für Lehrpersonen

- Personaldaten mit Privatadresse
- Logins und Berechtigungen

SOAP-Methode: `LO_getTeachers`
Parameter: `int semesterYear, int semesterPart`

Dateiname: `"SJ" + Schuljahr + "-" + Semester + "_Lehrpersonen.xml"`
Beispiel: `SJ2011-2_Lehrpersonen.xml`

1.3.2 Definitionen aller Schulklassen

- Klassendefinitionen
- Grunddaten der Schulklassen

SOAP-Methode: `LO_getSchoolclassDefinitions`
Parameter: `int semesterYear, int semesterPart`

Dateiname: `"SJ" + Schuljahr + "-" + Semester + "_Klassendefinitionen.xml"`
Beispiel: `SJ2011-2_Klassendefinitionen.xml`

1.3.3 Daten einer Schulklasse

- Schülerdaten mit Kontaktadressen
- Klassen- und Fachlehrpersonenzuteilungen
- Niveau- und Wahlfachzuteilungen

SOAP-Methode: `LO_getSchoolclass`
Parameter: `int semesterYear, int semesterPart, String classKey`

Alternative SOAP-Methode: `LO_getSchoolclass2`
Parameter: `int semesterYear, int semesterPart, String classKey, String classID`

Dateiname: `"SJ" + Schuljahr + "-" + Semester + "_Klasse_" + Klassenkürzel + ".xml"`
Beispiel: `SJ2011-2_Klasse_P1a.xml`

Es ist wichtig, dass die Daten für die Lehrpersonen zuerst für den Import zur Verfügung gestellt werden, da die Lehrpersonendaten für die Zuteilung dieser Lehrpersonen nachfolgend von den Klassendaten vorhanden sein müssen!

1.3.4 Definitionen aller Gruppen / Fachklassen

- Gruppendefinitionen
- Grunddaten der Gruppen

SOAP-Methode: `LO_getStudyGroupDefinitions`
Parameter: `int semesterYear, int semesterPart`

Dateiname: `"SJ" + Schuljahr + "-" + Semester + "_Gruppendefinitionen.xml"`

Beispiel: `2011-2_Gruppendefinitionen.xml`

1.3.5 Daten einer Arbeitsgruppe / Fachklasse

Schülerzuordnung
Lehrpersonenzuteilungen

SOAP-Methode: `LO_getStudyGroup`
Parameter: `int semesterYear, int semesterPart, String groupKey`

Alternative SOAP-Methode: `LO_getStudyGroup2`
Parameter: `int semesterYear, int semesterPart, String groupKey, String groupID`

Dateiname: `"SJ" + Schuljahr + "-" + Semester + "_Gruppe_" + Gruppenkürzel + ".xml"`

Beispiel: `SJ2011-2_Gruppe_EnglischE1.xml`

Lehrpersonen und die Schülerdaten müssen vorgängig importiert sein.

1.3.6 Weitere Prozesse

Künftige Implementationen könnten mit folgenden Methoden ausgebaut werden:

- Adressen von Behörden, Ärzten und Zahnärzten abholen mittels SOAP
SOAP-Methode: `LO_getAddresses`
- Dokumente wie Zeugnisse pro Schüler mittels SOAP hochladen
SOAP-Methode: `LO_sendDocs`

- Zeugniswerte der Schüler mittels SOAP hochladen
SOAP-Methode: LO_sendEndmarks

1.4 SOAP – Anforderung

Dieses Kapitel 'SOAP-Anforderung' kann beim dateibasierenden Export übersprungen werden. Der eigentliche Dateiinhalt wird ab Abschnitt 'Daten für Lehrpersonen – XML-Struktur' (siehe Seite 16) dokumentiert.

Transport-Methode: SOAP 1.2

Content-Type: application/soap+xml

<code><?xml></code>	<i>XML-Header</i>
<code><soap:Envelope></code>	<i>SOAP-Nachricht</i>
<code><soap:Header></code>	<i>Kopfbereich</i>
<code><ServiceAuthHeader></code>	<i>Authentifizierung beim Webservice</i>
<code></ServiceAuthHeader></code>	
<code></soap:Header></code>	
<code><soap:Body></code>	<i>Inhalt der Anfrage</i>
<code><LO_getSchoolclass></code>	<i>Methodenaufruf mit Parameter</i>
<code></LO_getSchoolclass></code>	
<code></soap:Body></code>	
<code></soap:Envelope></code>	

Auf nachfolgenden Seiten sind die einzelnen Elemente der SOAP-Anforderung dokumentiert.

```
<soap:Envelope>
```

SOAP-Nachricht

soap:Envelope

1.4.1 Element 'soap:Envelope'

zwingend notwendig

Dieses Element ist das eigentliche Root-Element der SOAP-Anforderung und darf nur einmal im XML enthalten sein. Dieses Element beinhaltet den SOAP-Header und den SOAP-Body.

Anzahl: 1

Beispiel:

```
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://www.w3.org/2003/05/soap-envelope">
  <soap:Header>
    ...
  </soap:Header>
  <soap:Body>
    ...
  </soap:Body>
</soap:Envelope>
```

1.4.2 Namespaces zum Element 'soap:Envelope'

`xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"`

zwingend notwendig

Der Namespace „xsi“ muss auf eine gültige XML-Schema Instanz URL verweisen. Wir empfehlen die XML-Schema Instanz des W3C beizubehalten.

Format: [URL](#)

Beispiel: <http://www.w3.org/2001/XMLSchema-instance>

`xmlns:xsd="http://www.w3.org/2001/XMLSchema"`

zwingend notwendig

Der Namespace „xsd“ muss auf eine gültige XML-Schema URL verweisen. Wir empfehlen das XML-Schema des W3C beizubehalten.

Format: [URL](#)

Beispiel: <http://www.w3.org/2001/XMLSchema>

`xmlns:soap=http://www.w3.org/2003/05/soap-envelope`

zwingend notwendig

Der Namespace „soap“ muss auf eine gültige SOAP-Envelope URL verweisen. Wir empfehlen den SOAP-Envelope Namespace des W3C beizubehalten (Version 1.2).

Format: [URL](#)

Beispiel: <http://www.w3.org/2003/05/soap-envelope>

<soap:Header>

Kopfbereich

soap:Envelope/soap:Header

1.4.3 Element 'soap:Header'

zwingend notwendig

Dieses Element beinhaltet die Kopfzeilen der SOAP-Anforderung, die nichts mit dem eigentlichen Inhalt zu tun haben, wie zum Beispiel Authentifizierungen. Der SOAP-Header muss im SOAP-Envelope vor dem SOAP-Body definiert werden.

Anzahl: 1

Beispiel:

```
<soap:Header>
  <ServiceAuthHeader
    xmlns="http://www.lehreroffice.ch/xmlns/soap/2010/exchange-v1">
    <username>LehrerOffice</username>
    <password>f62236383a6873329e955a1bf8353be7</password>
  </ServiceAuthHeader>
</soap:Header>
```

<ServiceAuthHeader>

Authentifizierung beim Webservice

soap:Envelope/soap:Header/ServiceAuthHeader

1.4.4 Element 'ServiceAuthHeader'

zwingend notwendig

Dieses Element beinhaltet die Authentifizierung der SOAP-Anforderung. Der ServiceAuthHeader muss im SOAP-Header definiert werden.

Hier wird der Namespace ohne Namespace-Präfix im ServiceAuthHeader-Tag angegeben und ist damit der Standardnamespace für alle Elemente innerhalb dieses Tags. Dadurch entfällt die Namespace-Zuweisung als Präfix vor den Elementnamen.

Anzahl: 1

Beispiel:

```
<ServiceAuthHeader xmlns="http://www.lehrer-  
office.ch/xmlns/soap/2010/exchange-v1">  
  <username>LehrerOffice</username>  
  <password>f62236383a6873329e955a1bf8353be7</password>  
</ServiceAuthHeader>
```

1.4.5 Attribute zum Element 'ServiceAuthHeader'

xmlns="http://www.lehreroffice.ch/xmlns/soap/2010/exchange-v1"

zwingend notwendig

Dieser Namespace muss als Standardnamespace, also ohne Namespace-Präfix angegeben werden.

Diese Versionsangabe hat keinen Zusammenhang mit der Programmversion von CMI LehrerOffice als Anwendung.

Format: URL

Beispiel: <http://www.lehreroffice.ch/xmlns/soap/2010/exchange-v1>

1.4.6 Elemente zu 'ServiceAuthHeader'

username

zwingend notwendig

Der Benutzername für die Authentifizierung beim Webservice.

Format: `alphanumerisch`

Beispiel: `Hans.Muster73`

password

zwingend notwendig

Das Passwort für die Authentifizierung beim Webservice als MD5- oder SHA-verschlüsselter String.

Format: `alphanumerisch`

Beispiel: `f62236383a6873329e955a1bf8353be7`

<soap:Body>

Inhalt der Anfrage

soap:Envelope/soap:Body

1.4.7 Element 'soap:Body'

zwingend notwendig

Dieses Element beinhaltet den Inhalt der SOAP-Anforderung, resp. die SOAP-Methode. Der SOAP-Body muss im SOAP-Envelope nach dem SOAP-Header definiert werden.

Die möglichen SOAP-Methoden und deren Parameter sind nachfolgend beschrieben.

Anzahl: 1

Komplettes Beispiel für eine SOAP-Anforderung:

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://www.w3.org/2003/05/soap-envelope">
  <soap:Header>
    <ServiceAuthHeader xmlns="http://www.lehreroffice.ch/xmlns/soap/2010/exchange-
v1">
      <username>LehrerOffice</username>
      <password>f62236383a6873329e955a1bf8353be7</password>
    </ServiceAuthHeader>
  </soap:Header>
  <soap:Body>
    <LO_getSchoolclass xmlns="http://www.lehreroffice.ch/xmlns/soap/2010/exchange-
v1">
      <semesterYear>2010</semesterYear>
      <semesterPart>1</semesterPart>
      <classKey>SHA-PS-1-b</classKey>
    </LO_getSchoolclass>
  </soap:Body>
</soap:Envelope>
```

1.5 SOAP – Antwort

Transport-Methode: SOAP 1.2

Content-Type: application/soap+xml

<code><?xml></code>	<i>XML-Header</i>
<code><soap:Envelope></code>	<i>SOAP-Nachricht</i>
<code><soap:Body></code>	<i>Inhalt der Antwort</i>
<code><LO_getSchoolclassResponse></code>	<i>Antwort von Methodenaufruf</i>
<code><LO_getSchoolclassResult></code>	<i>Resultat</i>
<code><lehreroffice></code>	<i>Dateninhalt für LehrerOffice</i>
<code></lehreroffice></code>	
<code></LO_getSchoolclassResult></code>	
<code></LO_getSchoolclassResponse></code>	
<code></soap:Body></code>	
<code></soap:Envelope></code>	

Nachfolgend sind die einzelnen Elemente der SOAP-Antwort dokumentiert.

```
<soap:Envelope>
```

SOAP-Nachricht

soap:Envelope

1.5.1 Element 'soap:Envelope'

zwingend notwendig

Dieses Element ist das eigentliche Root-Element der SOAP-Antwort und darf nur einmal im XML enthalten sein. Dieses Element beinhaltet den SOAP-Body.

Die SOAP-Antwort enthält in der Regel keinen SOAP-Header.

Anzahl: 1

Beispiel:

```
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://www.w3.org/2003/05/soap-envelope">
  <soap:Body>
    ...
  </soap:Body>
</soap:Envelope>
```

1.5.2 Namespaces zum Element 'soap:Envelope'

xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

zwingend notwendig

Der Namespace „xsi“ muss auf eine gültige XML-Schema Instanz URL verweisen. Wir empfehlen die XML-Schema Instanz des W3C beizubehalten.

Format: URL

Beispiel: <http://www.w3.org/2001/XMLSchema-instance>

xmlns:xsd="http://www.w3.org/2001/XMLSchema"

zwingend notwendig

Der Namespace „xsd“ muss auf eine gültige XML-Schema URL verweisen. Wir empfehlen das XML-Schema des W3C beizubehalten.

Format: URL

Beispiel: <http://www.w3.org/2001/XMLSchema>

xmlns:soap="http://www.w3.org/2003/05/soap-envelope"

zwingend notwendig

Der Namespace „soap“ muss auf eine gültige SOAP-Envelope URL verweisen. Wir empfehlen den SOAP-Envelope Namespace des W3C beizubehalten (Version 1.2).

Format: URL

Beispiel: <http://www.w3.org/2003/05/soap-envelope>

<soap:Body>	<i>Inhalt der Antwort</i>
-------------	---------------------------

soap:Envelope/soap:Body

1.5.3 Element 'soap:Body'

zwingend notwendig

Dieses Element beinhaltet den Inhalt der SOAP-Antwort, resp. das Ergebnis der SOAP-Anforderung. Die nächsten beiden Kinderelemente dieses Elementes werden immer nach der SOAP-Methode benannt zuzüglich der Postfixes „Response“ und „Result“.

Die möglichen SOAP-Methoden finden Sie in der Beschreibung der Prozesse.

Anzahl: 1

Beispiel:

```
<soap:Body>
  <LO_getSchoolclassResponse xmlns="http://www.lehrer-
office.ch/xmlns/soap/2010/exchange-v1">
    <LO_getSchoolclassResult>

      <lehreroffice ... >
        ... „Antwort der Anfrage als XML“
      </lehreroffice>

    </LO_getSchoolclassResult>
  </LO_getSchoolclassResponse>
</soap:Body>
```

Komplettes Beispiel für eine SOAP-Antwort:

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://www.w3.org/2003/05/soap-envelope">
  <soap:Body>
    <LO_getSchoolclassResponse xmlns="http://www.lehrer-
office.ch/xmlns/soap/2010/exchange-v1">
      <LO_getSchoolclassResult>

        <lehreroffice ... >
          ... „Antwort der Anfrage als XML“
        </lehreroffice>

      </LO_getSchoolclassResult>
    </LO_getSchoolclassResponse>
  </soap:Body>
</soap:Envelope>
```

1.6 SOAP – Fehler

Falls ein Fehler auftritt, kann entweder eine SOAP-Exception ausgelöst oder statt dem “lehreroffice”-Element ein Fehlerelement mit weiteren Informationen zur Fehlerbehandlung zurückgegeben werden.

1.6.1 Element 'LO_error'

zwingend notwendig

Dieses Element beinhaltet den Inhalt der Fehlermeldung.

Anzahl: 1

Beispiel:

```
<LO_Error>
  <errorCode>22</errorCode>
  <errorMessage>Klasse nicht gefunden</errorMessage>
  <errorDescription>Es wurde keine Klasse mit dem Schlüssel "1SEK" und/oder der ID
1123 in der Datenbank "lehreroffice" gefunden.</errorDescription>
</LO_Error>
```

1.6.2 Elemente zu 'LO_Error'

errorCode

zwingend notwendig

Gültiger Code des Fehlers gemäss Liste.

Werte: 10 = Anmeldung nicht erfolgreich
20 = Keine Daten zu diesen Parametern gefunden
21 = Semester nicht gefunden
22 = Klasse/Arbeitsgruppe nicht gefunden
99 = Anderer Fehler

Beispiel: 21

errorMessage

zwingend notwendig

Kurze Beschreibung des Fehlers als Fehlermeldung.

Format: alphanumerisch

Beispiel: Semester nicht gefunden

errorDescription

optional

Ausführliche Beschreibung des Fehlers.

Format: alphanumerisch

Beispiel: Keine gültige Jahresangabe: „2012/13“.

Komplettes Beispiel für eine SOAP-Antwort mit Fehlerelement:

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://www.w3.org/2003/05/soap-envelope">
  <soap:Body>
    <LO_getSchoolclassResponse xmlns="http://www.lehrer-
office.ch/xmlns/soap/2010/exchange-v1">
      <LO_getSchoolclassResult>

        <LO_Error>
          <errorCode>10</errorCode>
          <errorMessage>Anmeldung nicht erfolgreich</errorMessage>
          <errorDescription>Es sind nur Verbindungen über HTTPS erlaubt.</errorDescrip-
tion>
        </LO_Error>

      </LO_getSchoolclassResult>
    </LO_getSchoolclassResponse>
  </soap:Body>
</soap:Envelope>
```

2 Daten der Lehrpersonen

Mit dieser Methode können die Daten und Adressen der Lehrpersonendaten von der Schulverwaltung übernommen werden. Diese Angaben werden schlussendlich im Modul 'Adressen' von CMI LehrerOffice abgelegt werden (siehe Kategorie 'Lehrpersonen').

Daten für Lehrpersonen

- Personaldaten mit Privatadresse
- Logins und Berechtigungen

SOAP-Methode: `LO_getTeachers`

Parameter: `int semesterYear, int semesterPart`

Dateiname: `"SJ" + Schuljahr + "-" + Semester + "_Lehrpersonen.xml"`

Beispiel: `SJ2011-2_Lehrpersonen.xml`

2.1 SOAP – Anfrage für Daten der Lehrpersonen

```
<LO_getTeachers>
```

Methodenaufruf mit Parameter

```
soap:Envelope/soap:Body/LO_getTeachers
```

2.1.1 Element 'LO_getTeachers'

zwingend notwendig

Dieses Element ist der eigentliche Methodenaufruf für die Abfrage/Aktion der Lehrpersonen.

Jede Schulverwaltungslösung stellt diese SOAP-Methode als Web-Service zur Verfügung. Das erwartete Resultat als XML ist im Abschnitt 'Daten für Lehrpersonen – XML-Struktur' dokumentiert.

Weitere SOAP-Methoden und deren Parameter finden Sie nachfolgend beschrieben.

Beispiel:

```
<soap:Body>
  <LO_getTeachers xmlns="http://www.lehreroffice.ch/xmlns/soap/2010/exchange-v1">
    <semesterYear>2010</semesterYear>
    <semesterPart>1</semesterPart>
  </LO_getTeachers>
</soap:Body>
```

2.1.2 Attribute zum Element 'LO_getTeachers'

```
xmlns="http://www.lehreroffice.ch/xmlns/soap/2010/exchange-v1"
```

zwingend notwendig

Dieser Namespace muss als Standardnamespace, also ohne Namespace-Präfix angegeben werden.

Diese Versionsangabe hat keinen Zusammenhang mit der Programmversion von CMI LehrerOffice als Anwendung.

Format: [URL](#)

Beispiel: <http://www.lehreroffice.ch/xmlns/soap/2010/exchange-v1>

2.1.3 Elemente zu 'LO_getTeachers' als Parameter

```
semesterYear
```

zwingend notwendig

Vierstellige Ganzzahl für das Schuljahr, wobei das Kalenderjahr des Schulanfangs geliefert wird. Für das Schuljahr 2010/11 wird somit die Zahl '2010' geliefert.

Format: [numerisch \(4\)](#)

Beispiel: [2010](#)

semesterPart

zwingend notwendig

Einstellige Ganzzahl für die Semesterangabe.

Format: 1|2

Beispiel: 1

2.2 Daten für Lehrpersonen – XML-Struktur

XML-Tag	Dateninhalt
<lehreroffice>	Dateninhalt
<inst>	Institution / Schulgemeinde
<source>	Angaben zum Quellsystem
</source>	
<semesters>	Auflistung der Semester
<semester>	Semesterdaten
<teachers>	Auflistung aller Lehrpersonendaten
<teacher>	Adressdaten einer Lehrperson
<personalData>	Stammdaten der Lehrperson
<homeAddress>	Private Wohnadresse
</homeAddress>	
<freeFields>	Freifelder mit Zusatzinformationen
</freeFields>	
</personalData>	
<work>	Anstellung / Lehrpersoneninformationen
<workAddress>	Schuladresse
</workAddress>	
<activities>	Liste aller Tätigkeiten
<activity>	Tätigkeit / Aktivität
</activity>	
</activities>	
</work>	
<login>	Benutzerlogin
</login>	
</teacher>	
</teachers>	
</semester>	
</semesters>	
</inst>	
</lehreroffice>	

2.3 Daten für Lehrpersonen – Elemente

<lehreroffice>

Dateninhalt

Lehreroffice

2.3.1 Element 'lehreroffice'

zwingend notwendig

Dieses Element ist das eigentliche Root-Element der XML-Struktur und darf nur einmal im XML enthalten sein. Dieses Element wird bei einer SOAP-Rückgabe im Body geführt.

Anzahl: 1

Beispiel:

```
<lehreroffice version="2010.1" object="1"
xmlns="http://www.lehreroffice.ch/xmlns/lo/2010/exchange-v1">
...
</lehreroffice>
```

2.3.2 Namespaces zum Element 'lehreroffice'

xmlns="http://www.lehreroffice.ch/xmlns/lo/2010/exchange-v1"

zwingend notwendig

Der Standard-Namespace 'xmlns' muss auf einen gültigen CMI LehrerOffice – Namespace verweisen. Die Schema-Location liegt unter derselben URL mit dem Suffix „.xsd“. In diesem Schema sind alle XML-Definitionen hinterlegt und die XML-Dateien können dadurch auf ihre Korrektheit überprüft resp. validiert werden.

Diese Versionsangabe hat keinen Zusammenhang mit der Programmversion von CMI LehrerOffice als Anwendung.

Format: URL

Beispiel: <http://www.lehreroffice.ch/xmlns/lo/2010/exchange-v1>

2.3.3 Attribute zum Element 'lehreroffice'

version="2010.1"

zwingend notwendig

Das Attribut 'version' muss zwingend mit der Versionsangabe "2010.1" mitgeliefert werden. Diese Versionsnummer beschreibt das Format und die Struktur der vorliegenden XML-Daten. Die aktuelle Implementation wird nur XML-Dateien verarbeiten, welche diese Versionsnummer "2010.x" aufweisen.

Sobald die Struktur wesentlich verändert wird und die Abwärtskompatibilität nicht mehr gewährleistet ist, wird die Ganzzahl vor dem Punkt hochgezählt. Die Zahl nach dem Punkt zeichnen Unterversionen aus, welche noch abwärtskompatibel sind.

Diese Versionsangabe hat keinen Zusammenhang mit der Programmversion von CMI LehrerOffice als Anwendung.

Format: 9999.99

Beispiel: 2010.1

object="1"

zwingend notwendig

Zweck der Daten: 1 = Datenübertragung aus Schulverwaltung (Standard)

2 = Datenweitergabe bei Schulgemeindewechsel (Reserve)

Format: numerisch 1|2 (1)

target="xy"

optional

Umschreibt das Zielformat dieser XML-Struktur. Je nach Zielsystem wird aus Datenschutzgründen nur eine Teilmenge aller möglichen Daten geliefert.

Format: alphanumerisch

<inst>

Institution / Schulgemeinde

lehreroffice/inst

2.3.4 Element 'inst'

zwingend notwendig

Damit die Institution resp. Schulgemeinde der Daten eindeutig zugeordnet werden kann, ist dieses Element zwingend notwendig und darf nur einmal unterhalb vom Root-Element 'lehreroffice' enthalten sein.

Anzahl: 1

Beispiel:

```
<inst customerKey="2805" zip="8240" canton="SH" name="Thayngen">  
...  
</inst>
```

2.3.5 Attribute zum Element 'inst'

customerKey="2805"

zwingend notwendig

Eindeutige ID vom Quellsystem für diesen Mandanten als Kundennummer.

Damit das Einlesen von Daten einer falschen Schulgemeinde verhindert werden kann, sollte die Institution mittels dieses Attributs eindeutig definiert werden.

CMI LehrerOffice: Diese Angabe muss in CMI LehrerOffice unter 'Einstellungen>DB>Service-Schnittstelle>Kundennummer Fremdschlüssel' übereinstimmend erfasst werden, wenn eine Kunden- oder Mandantenkennung geliefert wird.

Format: `alphanumerisch`

Beispiel: 2805

canton="SH"

empfohlen

Kanton der Schulgemeinde als Kürzel. In Deutschland oder Österreich kann dieses Attribut leer geliefert oder weggelassen werden.

Format: `alphanumerisch (2)`

Beispiel: SH

name="Thayngen"

empfohlen

Name der Schulgemeinde. Angabe dient lediglich zur Information.

Format: `alphanumerisch`

Beispiel: Thayngen

zip="8240" veraltet

Postleitzahl der Schulgemeinde. Es sind maximal 5 Stellen erlaubt (Deutschland).

Wenn (noch) keine Kundennummer existiert oder lieferbar ist, wird die PLZ als Alternative zur Identifikation verwendet.

CMI LehrerOffice: Diese Angabe muss in CMI LehrerOffice unter 'Einstellungen>DB>Schule>Postleitzahl' übereinstimmend erfasst werden, wenn eine PLZ und keine Kundennummer geliefert wird.

Format: numerisch (5)

Beispiel: 8240

Sobald weder das Attribut „customerKey“ noch das Attribut „zip“ geliefert werden, wird der Import nicht durchgeführt, da das Risiko besteht, fremde Daten in eine falsche Datenbank zu übertragen.

<source>

Angaben zum Quellsystem

lehreroffice/inst/source

2.3.6 Element 'source'

zwingend notwendig

Diese Informationen dienen zur Identifikation des Quellsystems sprich der Schulverwaltungslösung. Diese Angaben werden lediglich als Information und allenfalls zur Fehlersuche verwendet.

Anzahl: 1

Beispiel:

```
<source>
  <appName>ScolarisWeb</appName>
  <appVersion>1.9</appVersion>
  <creationDate>2011-03-29</creationDate>
  <creationTime>14:02:59</creationTime>
</source>
```

2.3.7 Elemente zu 'source'

appName

zwingend notwendig

Name der eingesetzten Anwendung vom Quellsystem. Beliebiger String möglich.

Format: `alphanumerisch`

Beispiel: `WinSchule`

appVersion

zwingend notwendig

Versionsangabe der eingesetzten Anwendung vom Quellsystem. Beliebiger String möglich.

Format: `alphanumerisch`

Beispiel: `8.2`

creationDate

zwingend notwendig

Datum der Erstellung dieser XML-Daten.

Format: `YYYY-MM-DD`

Beispiel: `2011-03-29`

creationTime

zwingend notwendig

Zeitstempel der Erstellung dieser XML-Daten.

Format: `HH:MM:SS`

Beispiel: `14:02:59`

dbName

optional

Name der Datenbank resp. der Datenbankinstanz. Hiermit könnte z.B. eine Unterscheidung der produktiven Daten und der Testdatenbank erzielt werden.

Format: `alphanumerisch`

Beispiel: `Produktion`

`<semesters>` *Auflistung der Semester*

lehreroffice/inst/semesters

2.3.8 Element 'semesters'

zwingend notwendig

Dieses Element fungiert als Liste mit Unterelementen vom Typ '<semester>'.
Anzahl: 1

Beispiel:

```
<semesters>
  <semester>
    ...
  <semester>
</semesters>
```

`<semester>` *Semesterdaten*

lehreroffice/inst/semesters/semester

2.3.9 Element 'semester'

zwingend notwendig

Die Daten werden semesterabhängig abgefragt. Obwohl die Auflistung im Element 'semesters' mehrere Elemente von 'semester' enthalten kann, wird vorerst nur ein Semester pro XML-Datei unterstützt.

Anzahl: 1 (*später ausbaubar*)

Beispiel:

```
<semester year="2011" part="2">
  ...
</semester>
```

2.3.10 Attribute zum Element 'semester'

`year="2011"`

zwingend notwendig

Schuljahresangabe vom Semester, wobei das Kalenderjahr vom Schulanfang erwartet wird. Somit wird für das Schuljahr 2011/12 die Ganzzahl '2011' erwartet.

Format: *numerisch (4)*

Beispiel: 2011

`part="2"`

zwingend notwendig

Halbjahresangabe (Semester) als Ganzzahl.

1 = 1. Halbjahr

2 = 2. Halbjahr

Format: *numerisch 1|2 (1)*

Beispiel: 2

`<teachers>` *Auflistung aller Lehrpersonendaten*

lehreroffice/inst/semesters/semester/teachers

2.3.11 Element 'teachers'

zwingend notwendig

Dieses Element fungiert als Liste mit den entsprechenden Adresselementen vom Typ '<teacher>'. In der aktuellen Implementation ist lediglich ein solches Element erlaubt.

Anzahl: 1 (*später ausbaubar*)

Beispiel:

```
<teachers complete="1">  
  <teacher>  
    ...  
  <teacher>  
</teachers>
```

2.3.12 Attribute zum Element 'teachers'

`complete="1"`

zwingend notwendig

Wenn der Code "1" (Standard) übergeben wird, dann ist die nachfolgende Adressliste mit den Lehrpersonen vollständig. Dies bedeutet, dass alle Lehrpersonen enthalten sind. Überzählige Adressen innerhalb von CMI Lehrer-Office werden hiermit automatisch ausgeblendet und die dazugehörigen Logins entfernt.

Mit dem Code "0" werden die fehlenden Adressen hinzugefügt. Diese Option ist für das Nachliefern von einzelnen Adressen sinnvoll (Push-Variante).

- 0 = Adressliste ist nicht vollständig
- 1 = Adressliste ist komplett (Standard)

Format: *numerisch* 0|1 (1)

Beispiel: 1

<code><teacher></code>	<i>Adresdaten einer Lehrperson</i>
------------------------------	------------------------------------

lehreroffice/inst/semesters/semester/teachers/teacher

2.3.13 Element 'teacher'

wenn Daten vorhanden

Sämtliche Angaben einer Adresse werden in diesem Element abgespeichert, wobei pro Lehrperson ein solches Element notwendig ist.

Anzahl: 0..n

Beispiel:

```
<teacher ahv="756.0978.6543.21" key="1001"
      modDateTime="2010-01-15T14:32:45">
  ...
</teacher>
```

2.3.14 Attribute zum Element 'teacher'

ahv="756.0978.6543.21"

empfohlen

AHV-Nummer dient als Identifikator mit erster Priorität. Die Datenschnittstelle von CMI LehrerOffice sucht zuerst nach einem Datensatz mit gleicher AHV-Nummer. Wenn erfolgreich, wird dieser Datensatz mutiert. Sollte der Datensatz nicht gefunden werden, wird in zweiter Priorität nach dem Attribut 'key' gesucht (siehe nachfolgende Beschreibung).

Es wird die neue AHV-Nummer (AHVN13) mit 13 Stellen inklusive den 3 Punkten als Trennzeichen erwartet. Sind das Format und die Prüfsumme nicht korrekt, wird diese Adresse zurückgewiesen und nicht importiert.

Format: alphanumerisch (16)

Beispiel: 756.0978.6543.21

key="1001"

zwingend notwendig

Eindeutige ID vom Quellsystem für diesen Datensatz.

Wenn die Schulverwaltung noch keine gültigen AHV-Nummern besitzt kann in zweiter Priorität mit diesem Schlüsselwert abgeglichen werden. Sobald eine AHV-Nummer vorhanden ist, wird zuerst nach dieser gesucht. Wenn über AHV-Nummer selektiert und ein anderer Schlüsselwert überliefert wird, dann wird dieser innerhalb von CMI LehrerOffice geändert!

Format: alphanumerisch

Beispiel: 1001

id

optional

Eindeutige ID für die Lehrperson. Diese kann für weitere Systeme wie zB. NEST genutzt werden. Diese ID wird im CMI LehrerOffice als Fremdschlüssel gespeichert, aber nicht für die Identifikation der Lehrperson beim Datenimport genutzt.

Format: alphanumerisch

Beispiel: 98402

modDateTime

empfohlen

Modifikationsdatum im XSD-Format. Letzte Mutation des Datensatzes auf dem Quellsystem. Wenn dieser Zeitstempel älter oder gleich dem abgespeicherten TimeStamp in CMI LehrerOffice ist, wird dieses Element und alle Unterelemente nicht verarbeitet. Dank diesem TimeStamp kann die Verarbeitungszeit stark optimiert werden.

Auch bei einer Änderung der Aktivitäten, der Geschäftsadresse oder bei einem geänderten Loginname muss ein geänderter Zeitstempel geliefert werden.

Format: `YYYY-MM-DDTHH:MM:SS`

Beispiel: `2010-01-15T14:32:45`

action=""

(nicht implementiert) optional

Bearbeitungsaktion für Push-Variante, wenn die Mutationen laufend geliefert werden, denn hiermit müssen Löschaaktionen bewusst geliefert werden.

Standardmässig wird keine Aktion angegeben, da die überzähligen Adressen von CMI LehrerOffice ermittelt und so entfernt werden, sofern das Attribut 'complete' im Element 'teachers' mit dem Wert "1" gesetzt wurde.

Werte: `del` = Datensatz wird gelöscht (betrifft alle Semester)
`hide` = Datensatz wird ausgeblendet (betrifft alle Semester)

<personalData>

Stammdaten der Lehrperson

lehreroffice/inst/semesters/semester/teachers/teacher/personalData

2.3.15 Element 'personalData'

zwingend notwendig

Die eigentlichen Kontakt- und Personalangaben (Stammdaten) einer Lehrperson werden in diesem Element abgelegt.

Anzahl: 1

Beispiel:

```
<personalData>
  ...
</personalData>
```

2.3.16 Elemente zu 'personalData'

firstname

Vorname der Person.

Format: `alphanumerisch`

Beispiel: `Hans`

zwingend notwendig

surname

Nachname der Person.

Format: `alphanumerisch`

Beispiel: `Muster`

zwingend notwendig

gender

Geschlecht der Person.

Format: `alphanumerisch (1)`

Werte: `f` = weiblich (feminin)

`m` = männlich (masulin)

Beispiel: `m`

empfohlen

birthdate

Geburtsdatum.

Format: `YYYY-MM-DD`

Beispiel: `1973-03-22`

empfohlen

shortname

Initialen (Kürzel) der Lehrperson.

Format: `alphanumerisch (4)`

Beispiel: `hm`

empfohlen

aliasPrenome

Rufname des Vornamens der Person.

Format: `alphanumerisch`

Beispiel: `Hansli`

optional

aliasName optional

Rufname resp. Kurzform des Nachnamens. Einigen Schulen verwenden dieses Feld für die Kurzform des Namens bei langen ausländischen Namen.

Format: `alphanumerisch`

Beispiel: `Musterli`

nationality optional

Nationalität. Ausgeschriebene Landesangabe.

Format: `alphanumerisch`

Beispiel: `Deutschland`

nativeLanguage optional

Muttersprache der Lehrperson, wobei die Angabe als Nomen ausgeschrieben erwartet wird.

Format: `alphanumerisch`

Beispiel: `Spanisch`

profession (nicht implementiert) optional

Beruf

Format: `alphanumerisch`

Beispiel: `Primarlehrperson`

salutationAddress optional

Anrede für Anschrift

Format: `alphanumerisch`

Beispiel: `Herr`

salutationLetter optional

Briefanrede

Format: `alphanumerisch`

Beispiel: `Sehr geehrter Herr Muster`

<code><homeAddress></code>	<i>Private Wohnadresse</i>
----------------------------------	----------------------------

lehreroffice/inst/semesters/semester/teachers/teacher/personalData/homeAddress

2.3.17 Element 'homeAddress'

empfohlen

Private Adressangaben vom Wohnsitz der entsprechenden Person.

Anzahl: 1

Beispiel:

```
<homeAddress>
  <street>Hauptstrasse 16</street>
  <zip>8240</zip>
  <city>Thayngen</city>
  ...
</homeAddress>
```

2.3.18 Elemente zu 'homeAddress'

addressAddition optional

Zusätzliche Adressangabe mit c/o-Angabe, Flurname oder Liegenschaftsname (für die Kantone Thurgau und Appenzell Innerrhoden).

Format: `alphanumerisch`

Beispiel: `Hooptstross`

street empfohlen

Strassenangabe ohne Hausnummer. Es wird empfohlen die Strasse ausgeschrieben (nicht 'Hauptstr.' sondern mit 'Hauptstrasse' zu liefern).

Wenn in der Schulverwaltung die Angaben nicht getrennt vorliegen kann notfalls auch die Strassenangabe mit Hausnummer geliefert werden. Die Strassenangabe wird schlussendlich aus den 3 Teilen 'street', 'streetNumber' und 'streetNumberAddition' zusammengesetzt.

Format: `alphanumerisch`

Beispiel: `Hauptstrasse`

streetNumber empfohlen

Hausnummer der Strassenangabe.

Format: `numerisch (4)`

Beispiel: `16`

streetNumberAddition optional

Zusatzangabe resp. Hausteil zur Hausnummer.

Format: `alphanumerisch`

Beispiel: `b`

poBox optional

Postfachangabe, sofern vorhanden.

Format: `alphanumerisch`

Beispiel: `Postfach 776`

zip empfohlen
Postleitzahl. Es sind maximal 5 Stellen erlaubt (Deutschland).
Format: `numerisch (5)`
Beispiel: `8240`

city empfohlen
Ortsangabe des Wohnorts.
Format: `alphanumerisch`
Beispiel: `Thayngen`

canton optional
Wohnkanton.
Format: `alphanumerisch (2)`
Beispiel: `SH`

country optional
Angabe zum Land.
Format: `alphanumerisch`
Beispiel: `Schweiz`

phone empfohlen
Private Telefonnummer vom Festnetzanschluss.
Format: `alphanumerisch`
Beispiel: `052 720 13 29`

mobile empfohlen
Private Mobil-Nummer (Natel).
Format: `alphanumerisch`
Beispiel: `078 323 13 29`

fax optional
Private Faxnummer.
Format: `alphanumerisch`
Beispiel: `052 440 10 21`

email empfohlen
Private E-Mailadresse.
Format: `alphanumerisch`
Beispiel: `hmuster@bluewin.ch`

website optional
Vollständige URL der privaten Homepage.
Format: `alphanumerisch`
Beispiel: `http://www.meinewebsite.ch`

<code><freeFields></code>	<i>Freifelder mit Zusatzinformationen</i>
---------------------------------	---

lehreroffice/inst/semesters/semester/teachers/teacher/personalData/freeFields

2.3.19 Element 'freeFields'

optional

In CMI LehrerOffice können bei den Adressenstammdaten 3 Freifelder von der Schule frei definiert und so beliebige Zusatzangaben erfasst werden. Unter Umständen wird die Zahl der Freifelder in CMI LehrerOffice ausgebaut werden, deshalb kann bei der XML-Schnittstelle bereits eine Liste mit unbeschränkt vielen Feldern geliefert werden:

Anzahl: 1

Beispiel:

```
<freeFields>
  <freeField key="1" caption="KV">Helsana</freeField>
  <freeField key="2" caption="Nummer">12345678</freeField>
  ...
</freeFields>
```

2.3.20 Elemente zu 'freeFields'

freeField

optional

Jedes einzelne Freifeld wird als reines Textfeld geliefert und so im entsprechenden Freifeld in CMI LehrerOffice abgelegt. Die Zuordnung erfolgt über das Attribut "key". Mit key="2" wird zum Beispiel das zweite Freifeld definiert. Die Bezeichnung des Inhalts wird über das Attribut "caption" gesteuert und muss im CMI LehrerOffice vorerst manuell konfiguriert werden.

Wenn der als Textwert ein Leerstring geliefert wird, dann wird die allenfalls vorhandene Eingabe in CMI LehrerOffice gelöscht resp. mit dem Leerwert überschrieben werden!

Format: *alphanumerisch*

Beispiel: 12345678

<code><work></code>	<i>Anstellung / Lehrpersoneninformationen</i>
---------------------------	---

lehreroffice/inst/semesters/semester/teachers/teacher/work

2.3.21 Element 'work'

empfohlen

Adressangaben der Behörde resp. der Schule. Zusätzlich sind die Anstellungsinformationen in diesem Element enthalten.

Anzahl: 1

Beispiel:

```
<work>
  <room>EG 1</room>
  ...
</work>
```

2.3.22 Elemente zu 'work'

entranceDate optional

Eintrittsdatum / Anstellungsdatum

Format: YYYY-MM-DD

Beispiel: 2001-07-30

leavingDate optional

Austrittsdatum der Arbeitsstelle

Format: YYYY-MM-DD

Beispiel: 2011-07-30

room optional

Nummer des Schulzimmers / Klassenzimmers.

Format: alphanumerisch

Beispiel: A21

schoolYears optional

Anzugeben ist die Anzahl Jahre, während der die Person als Lehrkraft im Kantonsgebiet beschäftigt ist (Dabei werden auch Jahre gezählt, in denen die Lehrkraft in Teilzeitbeschäftigung gearbeitet hat). Diese Angabe entspricht dem Wert 'Jahre im Schuldienst' (B.3) im Feld 'yearsInAct' vom Bundesamt für Statistik.

Format: numerisch

Beispiel: 2

mainFunction optional

Haupttätigkeit der Lehrperson. Unter <activities> können weitere Funktionen z.B. pro Schuleinheit definiert wird.

Format: alphanumerisch

Beispiel: Werklehrperson

personalNumber optional

Personalnummer der Lehrperson.

Format: alphanumerisch

Beispiel: 12345

<code><workAddress></code>	<i>Schuladresse</i>
----------------------------------	---------------------

lehreroffice/inst/semesters/semester/teachers/teacher/work/workAddress

2.3.23 Element 'workAddress'

empfohlen

Adressangaben der Behörde resp. der Schule. Zusätzlich sind die Anstellungsinformationen in diesem Element enthalten.

Anzahl: 1

Beispiel:

```
<workAddress>
  <instName>Schule Thayngen</instName>
  <instAddition>Oberstufenzentrum</instAddition>
  <street>Schulweg</street>
  <zip>8240</zip>
  <city>Thayngen</city>
  ...
</workAddress>
```

2.3.24 Elemente zu 'workAddress'

instName

empfohlen

Name der Institution der Behörde resp. der Schule

Format: *alphanumerisch*

Beispiel: *Schule Thayngen*

instAddition

optional

Zusatzangabe der Institution

Format: *alphanumerisch*

Beispiel: *Oberstufenzentrum*

addressAddition

optional

Zusätzliche Adressangabe mit c/o-Angabe, Flurname oder Liegenschaftsname (für die Kantone Thurgau und Appenzell Innerrhoden).

Format: *alphanumerisch*

Beispiel: *im Rütli*

street

empfohlen

Strassenangabe ohne Hausnummer. Es wird empfohlen die Strasse ausgeschrieben (nicht 'Hauptstr.' sondern mit 'Hauptstrasse' zu liefern).

Wenn in der Schulverwaltung die Angaben nicht getrennt vorliegen kann notfalls auch die Strassenangabe mit Hausnummer geliefert werden. Die Strassenangabe wird schlussendlich aus den 3 Teilen 'street', 'streetNumber' und 'streetNumberAddition' zusammengesetzt.

Format: *alphanumerisch*

Beispiel: *Schulweg*

<p>streetNumber Hausnummer der Strassenangabe. Format: <code>numerisch (4)</code> Beispiel: <code>2</code></p>	<p>empfohlen</p>
<p>streetNumberAddition Zusatzangabe resp. Hausteil zur Hausnummer. Format: <code>alphanumerisch</code> Beispiel: <code>b</code></p>	<p>optional</p>
<p>poBox Postfachangabe, sofern vorhanden. Format: <code>alphanumerisch</code> Beispiel: <code>Postfach 12</code></p>	<p>optional</p>
<p>zip Postleitzahl der Behörde resp. der Schulgemeinde. Es sind maximal 5 Stellen erlaubt (Deutschland). Format: <code>numerisch (5)</code> Beispiel: <code>8240</code></p>	<p>empfohlen</p>
<p>city Ortsangabe des Geschäfts resp. der Schulgemeinde. Format: <code>alphanumerisch</code> Beispiel: <code>Thayngen</code></p>	<p>empfohlen</p>
<p>canton Kanton des Arbeitgebers. Format: <code>alphanumerisch (2)</code> Beispiel: <code>SH</code></p>	<p>optional</p>
<p>country Angabe zum Land. Format: <code>alphanumerisch</code> Beispiel: <code>Schweiz</code></p>	<p>optional</p>
<p>phone Persönliche Telefonnummer (Direktwahl). Format: <code>alphanumerisch</code> Beispiel: <code>052 440 10 29</code></p>	<p>empfohlen</p>
<p>phoneMain Haupttelefonnummer (Zentrale oder vom Sekretariat). Format: <code>alphanumerisch</code> Beispiel: <code>052 440 10 20</code></p>	<p>optional</p>
<p>mobile Mobil-Nummer (Natel) der Schule. Die private Mobil-Nummer ist hingegen unter <home> zu liefern. Format: <code>alphanumerisch</code> Beispiel: <code>078 443 14 49</code></p>	<p>optional</p>

fax

veraltet

Faxnummer.

Format: [alphanumerisch](#)

Beispiel: [052 440 10 21](#)

email

empfohlen

Geschäftliche E-Mailadresse.

Format: [alphanumerisch](#)

Beispiel: [hans.muster@unsereschule.ch](#)

emailSchool

optional

Für eine zweite geschäftliche E-Mailadresse. Beide E-Mailadressen werden in CMI LehrerOffice ins gleiche Feld eingetragen. Sofern die Daten in 'email' und 'emailSchool' gleich sind, wird die E-Mailadresse nur einmal eingetragen.

Format: [alphanumerisch](#)

Beispiel: [hans.muster@unsereschule.ch](#)

website

optional

Vollständige URL der Homepage der Schule.

Format: [alphanumerisch](#)

Beispiel: [http://www.schulexy.ch/hmuster](#)

<code><activities></code>	<i>Liste aller Tätigkeiten</i>
---------------------------------	--------------------------------

lehreroffice/inst/semesters/semester/teachers/teacher/work/activities

2.3.25 Element 'activities'

optional

Dieses Element fungiert als Liste mit den Tätigkeiten resp. Aktivitäten einer Lehrperson vom Typ '<activity>'. Die Tätigkeiten können in der aktuellen Programmversion 2013.7 von CMI LehrerOffice noch nicht verwaltet werden. Die Tätigkeiten können aber trotzdem bereits geliefert werden. Wichtig ist, dass die Haupttätigkeit an erster Stelle geliefert wird.

Anzahl: 1

Beispiel:

```
<activities complete="1">  
  <activity key="1">  
  </activity>  
</activities>
```

2.3.26 Attribute zum Element 'activities'

`complete="1"`

zwingend notwendig

Wenn der Code "1" (Standard) übergeben wird, dann ist die nachfolgende Liste mit den Tätigkeiten vollständig. Dies bedeutet, dass alle Aktivitäten, womit die Lehrperson tätig ist, enthalten sind. Überzählige Tätigkeiten innerhalb von CMI LehrerOffice werden hiermit automatisch entfernt.

Mit dem Code "0" werden die fehlenden Tätigkeiten hinzugefügt. Diese Option ist für das Nachliefern von einzelnen Tätigkeiten sinnvoll (Push-Variante).

- 0 = Liste mit Tätigkeiten ist nicht vollständig
- 1 = Liste mit Tätigkeiten ist komplett (Standard)

Format: `numerisch 0|1 (1)`

Beispiel: 1

<code><activity></code>	<i>Tätigkeit / Aktivität</i>
-------------------------------	------------------------------

lehreroffice/inst/semesters/semester/teachers/teacher/workactivities/activity

2.3.27 Element 'activity'

wenn Daten vorhanden

Einzelne Lehrpersonen können mehrere Tätigkeiten innerhalb einer Schulgemeinde wahrnehmen. So kann ein Schulleiter zusätzlich die Aktivität als Fachlehrperson besitzen. Bei grossen Schulgemeinden kann zudem die Tätigkeiten in mehreren Schuleinheiten unterteilt werden. Es ist wichtig, dass die Hauptaktivität an erster Stelle geliefert wird.

Anzahl: `0..n`

Beispiel:

```
<activity key="1">  
  <unitKey>OZ1</unitKey>  
</activity>
```

2.3.28 Attribute zum Element 'activity'

`key="1"`

optional

Eindeutige ID vom Quellsystem für die Tätigkeit. Dies kann auch eine fortlaufende Tätigkeitsnummer für die entsprechende Lehrperson sein. Dieser Key muss somit nur innerhalb dieser Person eindeutig sein.

Format: `alphanumerisch`

Beispiel: `1`

`action=""`

optional

Bearbeitungsaktion für Push-Variante, wenn die Mutationen laufend geliefert werden, denn hiermit müssen Löschaaktionen bewusst geliefert werden.

Standardmässig wird keine Aktion angegeben, da die überzähligen Tätigkeiten von CMI LehrerOffice ermittelt und so entfernt werden, sofern das Attribut 'complete' im Element 'activities' mit dem Wert "1" gesetzt wurde.

Werte: `del` = Tätigkeit wird entfernt

2.3.29 Elemente zu 'activity'

`unitKey`

empfohlen

Grosse Schulgemeinden können in Schuleinheiten unterteilt werden, wobei jede Tätigkeit einer Schuleinheit zugeordnet werden kann.

Eindeutige ID vom Quellsystem für die Schuleinheit. Die Schuleinheiten müssen vorab innerhalb von CMI LehrerOffice definiert und mit demselben Schlüsselwert wie im Quellsystem eingestellt werden.

Format: `alphanumerisch`

Beispiel: `OZ1`

`unitName`

empfohlen

Name der Schuleinheit. Dient lediglich als Information für die Zuordnung der Schuleinheit zur ID, wird nicht übernommen.

Format: `alphanumerisch`

Beispiel: `Sekundarschule`

bur="72765790"

empfohlen

Zur Identifikation wird die BUR-Nummer SSP vom Amt für Statistik verwendet. Diese BUR-Nummer SSP ist ein eindeutiger Identifikator der Bildungsinstitute der Schweiz.

Format: [numerisch \(20\)](#)

Beispiel: [72765790](#)

function

empfohlen

Beschreibung der Tätigkeit/Funktion.

Format: [alphanumerisch](#)

Beispiel: [Werklehrer](#)

functionCategory

optional

Kategorie der Tätigkeit.

Werte: [LP](#) = Lehrperson

[SL](#) = Schulleitungspersonal

[SHP](#) = Sonderpädagogisches / Schulheilpädagogisches Personal

Beispiel: [LP](#)

employment

optional

Die Arbeitsverträge werden nach der vorgesehenen Einstellungsdauer unterschieden. Der unbefristete Arbeitsvertrag wird zwischen Arbeitgeber und Arbeitnehmer geschlossen und enthält keine Angaben zum Vertragsende. Als befristet gilt ein Arbeitsvertrag dann, wenn Arbeitgeber und Arbeitnehmer sich über ein Vertragsende einig sind, das an objektive Bedingungen geknüpft ist.

Dieser Wert entspricht dem 'Art des Arbeitsvertrags' (C.3) vom Feld 'status' vom Bundesamt für Statistik.

Werte: [1](#) = Unbefristeter Arbeitsvertrag

[2](#) = Befristeter Arbeitsvertrag

Beispiel: [1](#)

qualification

optional

Es handelt sich dabei um erlangte Qualifikationen, die für die erhobene Tätigkeit relevant sind. Der Zweck und die Nomenklatur wurden vom Bundesamt für Statistik übernommen (siehe Diplom / Qualifikationen C.4).

Werte / Nomenklatur der Qualifikationen der Lehrkräfte:

[11](#) = Lehrdiplom, das den Voraussetzungen entspricht, die in den kantonalen Anstellungsbedingungen der entsprechenden Schulart definiert sind.

[12](#) = Lehrdiplom, das teilweise den Voraussetzungen entspricht, die in den kantonalen Anstellungsbedingungen der Schulart definiert sind

[13](#) = Lehrdiplom, das den Voraussetzungen, die in den kantonalen Anstellungsbedingungen der entsprechenden Schulart definiert sind, nicht entspricht

[14](#) = Kein Lehrdiplom

[15](#) = Ausländisches Lehrdiplom für ausländische Schulen

Werte / Nomenklatur der Qualifikationen des Schulleitungspersonals:

[21](#) = Lehrdiplom

[22](#) = Spezifisches Schulleitungsdiplom

[23](#) = Allgemeines Managementdiplom (z.B. Führung, Verwaltung), das nicht auf die Schulleitungstätigkeit spezialisiert ist

[24](#) = Anderes Diplom

Format: [numerisch](#)

Beispiel: [11](#)

schoolForm

optional

Mit Schulart werden die nach inhaltlichen Kriterien unterschiedenen Ausbildungsgänge bezeichnet. Die Werte entsprechen der Liste wie beim Element 'classForm' vom Abschnitt 'classDefinition'.

Werte:	KG	=	Kindergarten
	BS	=	Basisstufe
	ES	=	Einschulungsklasse
	KK[n]	=	Kleinklasse A..D
	O	=	Sonderklasse
	P	=	Primarklasse
	S[n]	=	Sekundarklasse (Abteilung, Niveau als zweites Zeichen z.B. "SA")
	R	=	Realklasse
	W	=	Werkklasse

Beispiel: S

workload

optional

Pensum mit Anzahl Wochenstunden, die für eine Tätigkeit eingesetzt werden. Als Attribut 'fulltimeRef' kann die Anzahl Wochenstunden (Standard: 42 Stunden), die einer Vollzeitstelle für eine Tätigkeit entsprechen, als Information mitgeliefert werden.

Format: numerisch (5.2)

Beispiel: `<workload fulltimeRef="40.5">20</workload>`

<code><login></code>	<i>Benutzerlogin</i>
----------------------------	----------------------

lehreroffice/inst/semesters/semester/teachers/teacher/login

2.3.30 Element 'login'

optional

Damit die Benutzerlogins ebenfalls vom Quellsystem verwaltet werden können, kann ein Benutzerlogin für CMI LehrerOffice über dieses Element automatisiert erstellt und gepflegt werden.

Anzahl: 1

Beispiel:

```
<login username="hans.muster">
</login>
```

2.3.31 Attribute zum Element 'login'

username="hans.muster"

zwingend notwendig

Jedes Login muss einen eindeutigen Benutzernamen umfassen. Sollte das Login mit dem gelieferten Benutzernamen fehlen, wird das Login automatisch erstellt und mit der dazugehörigen Adresse in CMI LehrerOffice verknüpft.

Sollte ein Login mit dem gelieferten Benutzernamen gefunden werden, aber mit einer anderen Adresse bereits verknüpft sein, wird dieser Prozess mit den Benutzerdaten abgebrochen.

Der Benutzernamen darf keine Leerzeichen und Umlaute wie 'ä', 'ö', 'ü' oder 'é' enthalten. Zudem sind Hochkommas und andere Sonderzeichen nicht erlaubt.

Format: alphanumerisch

Beispiel: hans.muster

action=""

optional

Bearbeitungsaktion für Push-Variante, wenn die Mutationen laufend geliefert werden, denn hiermit müssen Löschaaktionen bewusst geliefert werden.

Werte: `del` = Benutzerlogin mit obigem Benutzernamen entfernen
(dazugehöriger Adressdatensatz wird nicht gelöscht)
`lock` = Benutzerlogin mit obigem Benutzernamen sperren
`unlock` = Benutzerlogin mit obigem Benutzernamen entsperren

2.3.32 Elemente zu 'login'

template optional

Benutzerrechte gemäss Vorlage eines bestehenden Benutzerkontos übertragen. Wenn kein Template angegeben wurde und das Login neu erstellt wurde, werden die Standardrechte für Klassen- und Fachlehrpersonen gesetzt.

Format: `alphanumerisch`

Beispiel: `vorlage.klassenlehrperson`

changeUsername (nicht implementiert) optional

Existierenden Benutzernamen in der CMI LehrerOffice-Datenbank umbenennen. Bitte die Hinweise beim Attribut 'username' beachten. Diese Aktion wird notwendig, wenn der Benutzername zum Beispiel nach einer Heirat geändert werden muss.

Format: `alphanumerisch`

Beispiel: `hans.muster-frei`

3 Definitionen aller Schulklassen

Mit dieser Methode können die Schulklassendefinitionen von der Schulverwaltung übernommen werden. Diese Angaben werden schlussendlich im Modul 'Klassen' von CMI LehrerOffice abgelegt werden.

Definitionen aller Schulklassen

- Klassendefinitionen
- Grunddaten der Schulklassen

SOAP-Methode: `LO_getSchoolclassDefinitions`

Parameter: `int semesterYear, int semesterPart`

Dateiname: `"SJ" + Schuljahr + "-" + Semester + "_Klassendefinitionen.xml"`

Beispiel: `SJ2011-2_Klassendefinitionen.xml`

3.1 SOAP – Anfrage für die Definitionen aller Schulklassen

```
<LO_getSchoolclassDefinitions>
```

Methodenaufruf mit Parameter

```
soap:Envelope/soap:Body/LO_getSchoolclassDefinitions
```

3.1.1 Element 'LO_getSchoolclassDefinitions'

zwingend notwendig

Dieses Element ist der eigentliche Methodenaufruf für die Abfrage/Aktion der Klassendefinitionen.

Jede Schulverwaltungslösung stellt diese SOAP-Methode als Web-Service zur Verfügung. Das erwartete Resultat als XML ist im Abschnitt 'Definitionen aller Schulklassen – XML-Struktur' dokumentiert.

Weitere SOAP-Methoden und deren Parameter finden Sie nachfolgend beschrieben.

Beispiel:

```
<soap:Body>
  <LO_getSchoolclassDefinitions xmlns="http://www.lehrer-
office.ch/xmlns/soap/2010/exchange-v1">
    <semesterYear>2010</semesterYear>
    <semesterPart>1</semesterPart>
  </LO_getSchoolclassDefinitions>
</soap:Body>
```

3.1.2 Attribute zum Element 'LO_getSchoolclassDefinitions'

```
xmlns="http://www.lehreroffice.ch/xmlns/soap/2010/exchange-v1"
```

zwingend notwendig

Dieser Namespace muss als Standardnamespace, also ohne Namespace-Präfix angegeben werden.

Diese Versionsangabe hat keinen Zusammenhang mit der Programmversion von CMI LehrerOffice als Anwendung.

Format: URL

Beispiel: <http://www.lehreroffice.ch/xmlns/soap/2010/exchange-v1>

3.1.3 Elemente zu 'LO_getSchoolclassDefinitions' als Parameter

`semesterYear`

zwingend notwendig

Vierstellige Ganzzahl für das Schuljahr, wobei das Kalenderjahr des Schulanfangs geliefert wird. Für das Schuljahr 2010/11 wird somit die Zahl '2010' geliefert.

Format: `numerisch (4)`

Beispiel: `2010`

`semesterPart`

zwingend notwendig

Einstellige Ganzzahl für die Semesterangabe.

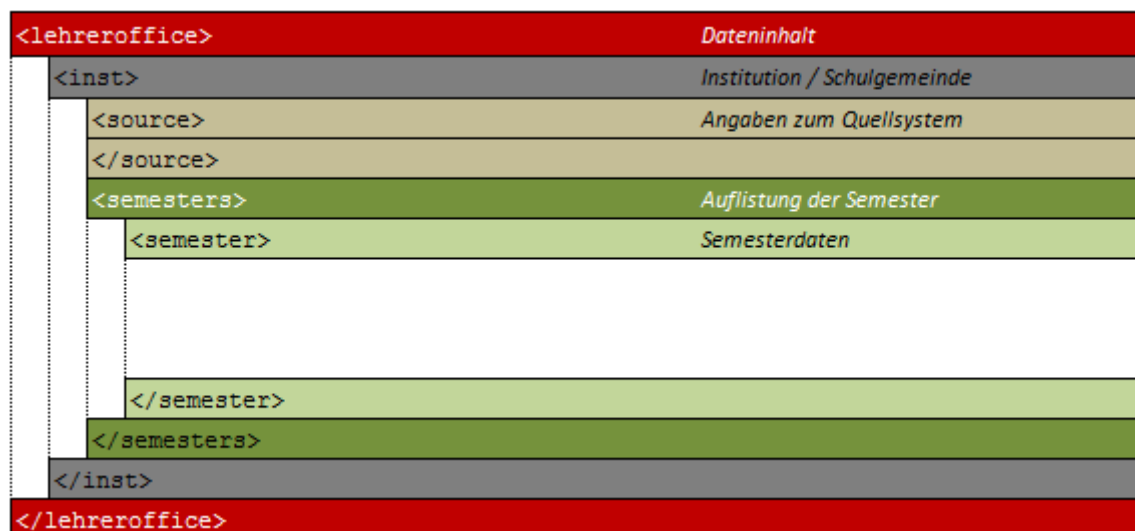
Format: `1|2`

Beispiel: `1`

3.2 Definitionen aller Schulklassen – XML-Struktur



3.3 Definitionen aller Schulklassen – Elemente



Obige Elemente und dessen Attribute und Unterelemente sind im vorangehenden Kapitel 'Daten für Lehrpersonen – Elemente' detailliert dokumentiert. Nachfolgend sind nun die Elemente innerhalb vom Element '<semester>' beschrieben.

`<schoolclasses>` *Auflistung der Schulklassen*

lehreroffice/inst/semesters/semester/schoolclasses

3.3.1 Element 'schoolclasses'

zwingend notwendig

Dieses Element fungiert als Liste mit Unterelementen vom Typ '`<schoolclass>`'. Hier müssen alle Schulklassen geliefert werden.

Anzahl: 1

Beispiel:

```
<schoolclasses complete="1">
  <schoolclass key="1A">
    ...
  </schoolclass>
  <schoolclass key="1B">
    ...
  </schoolclass>
  ...
</schoolclasses>
```

3.3.2 Attribute zum Element 'schoolclasses'

`complete="1"`

zwingend notwendig

Wenn der Code "1" übergeben wird, dann ist die nachfolgende Liste mit den Schulklassen vollständig. Dies bedeutet, dass alle Schulklassen enthalten sind. Überzählige Schulklassen werden hiermit automatisch innerhalb CMI LehrerOffice ausgeblendet.

Mit dem Code "0" werden die fehlenden Schulklassen hinzugefügt. Diese Option ist für das Nachliefern von einzelnen Gruppen sinnvoll (Push-Variante).

- 0 = Liste mit Schulklassen ist nicht vollständig
- 1 = Liste mit Schulklassen ist komplett

Format: `numerisch 0|1 (1)`

Beispiel: 1

<code><schoolclass></code>	<i>Schulklasse</i>
----------------------------------	--------------------

lehreroffice/inst/semesters/semester/schoolclasses/schoolclass

3.3.3 Element 'schoolclass'

zwingend notwendig

Hier müssen alle Schulklassen geliefert werden.

Anzahl: 1..n

Beispiel:

```
<schoolclass key="SHA-PS-1-b" id="9" active="1" modDateTime="2010-01-03T09:32:59">
  ...
</schoolclass>
```

3.3.4 Attribute zum Element 'schoolclass'

key="SHA-PS-1-b"

zwingend notwendig

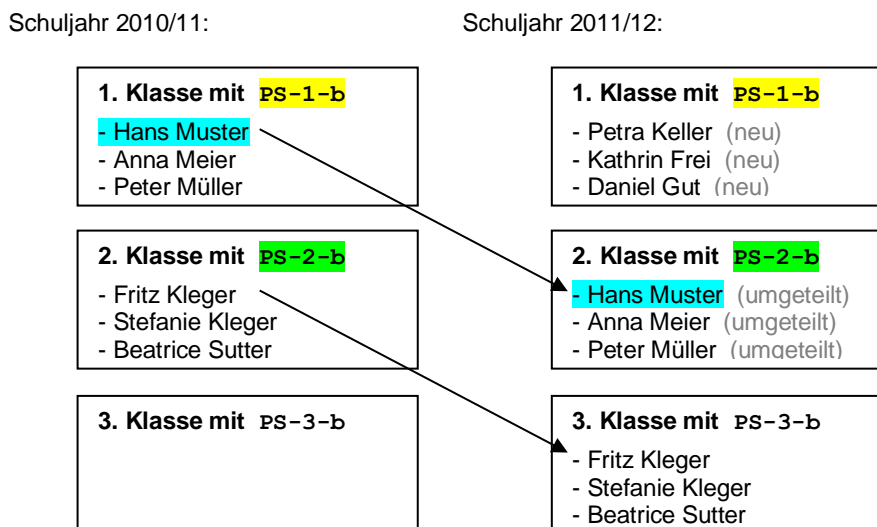
Eindeutige und statische ID vom Quellsystem für die Schulklasse, welche über alle Schuljahre z.B. für die 1. Primar-klasse unverändert bleibt.

Format: alphanumerisch

Beispiel: SHA-PS-1-b

In CMI LehrerOffice werden die Klassengefässe einmalig definiert, wobei dazu für jede Klasse die dazugehörigen Schulfächer, Lernziele und allenfalls Kompetenzraster hinterlegt werden. Die einzelnen Schüler und die entsprechenden Klassenlehrer werden am Ende eines Schuljahres nun von der aktuellen Klassendefinition zur nächsten weitergeschoben. Vergleichbar mit einem Klassenzimmer, das immer wieder mit neuen Schülern und Lehrpersonen belegt wird.

Die Klassendefinitionen mit ihren internen ID's bleiben somit statisch und für jedes Schuljahr grundsätzlich identisch. Siehe nachfolgendes Beispiel:



Diese ID kann selbstverständlich auch aus verschiedenen Feldern zusammengestellt werden, wobei eine eindeutige ID sichergestellt werden muss. Ein möglicher Ansatz ist:

Schuleinheit + "-" + Stufe + "-" + Klasse (Integer) + "-" + Gruppe

Beispiele: SHA-PS-2-b (Schulhaus Amden, Primarschule, 2. Klasse b)
OZ1-REAL-1-a (Oberstufenzentrum 1, Real, 1. Klasse a)

Der Schlüsselwert muss in CMI LehrerOffice gemäss Klassenliste der Schulverwaltungslösung im Modul 'Klassen' im Feld 'Schlüssel' einmalig erfasst werden, wobei die Gross- und Kleinschreibung nicht berücksichtigt wird. Ein direkter Import mit automatischer Erstellung von neuen Schulklassen in CMI LehrerOffice ist vorerst nicht geplant, da die Fachdefinitionen und Kompetenzraster entsprechend zugewiesen werden müssen.

Wenn die Schulklasse mit dem identischen Key nicht in CMI LehrerOffice gefunden wird, dann werden alle nachfolgenden Schülerdaten dieser Schulklasse nicht importiert.

id optional

Bei Schulverwaltungslösungen (wie SclarisWeb), welche für jede Schulklasse eine neue ID zu Beginn eines Schuljahres generieren, kann die ID für ein Schuljahr in CMI LehrerOffice auch als zweites Suchkriterium genutzt werden.

Dies bedeutet, dass z.B. bei einer 1. Primarklasse die ID mit "9" in CMI LehrerOffice als Fremdschlüssel bei der entsprechenden 1. Klasse für ein Schuljahr geführt wird.

Schulverwaltungslösungen (wie iCampus), welche dasselbe System mit den fixen Schlüsselwerten wie CMI LehrerOffice anwendet, darf dieses Attribut nicht nutzen.

Format: `alphanumerisch`

Beispiel: 9

modDateTime optional

Modifikationsdatum im XSD-Format. Letzte Mutation des Datensatzes auf dem Quellsystem. Wenn dieser Zeitstempel älter oder gleich dem abgespeicherten TimeStamp in CMI LehrerOffice ist, wird dieses Element und alle Unterelemente nicht verarbeitet. Dank diesem TimeStamp kann die Verarbeitungszeit stark optimiert werden.

Auch bei einer Änderung der dazugehörigen Klassen- oder Fachlehrpersonen, der untergeordneten Schülerdaten oder Fachzuteilungen muss ein geänderter Zeitstempel geliefert werden!

Format: `YYYY-MM-DDTHH:MM:SS`

Beispiel: `2010-01-03T09:32:59`

active optional

Schulklasse ist in diesem Semester aktiv (1) gekennzeichnet oder nicht (0).

Format: `numerisch 0|1 (1)`

Beispiel: 1

<classDefinition>

Klassendefinition

lehreroffice/inst/semesters/semester/schoolclasses/schoolclass/classDefinition

3.3.5 Element 'classDefinition'

optional

Damit Klassengefässe automatisch in CMI LehrerOffice erstellt werden können, kann die Definition der Klasse mit diesen Angaben einmalig erstellt werden, wenn die Klasse gemäss Schlüssel nicht gefunden werden konnte. Diese Definitionen sind im Gegensatz zu 'classData' nicht semesterabhängig.

Anzahl: 1

Beispiel:

```
<classDefinition>
  ...
</classDefinition>
```

3.3.6 Attribute zu 'classDefinition'

modDateTime

optional

Modifikationsdatum im XSD-Format. Letzte Mutation des Datensatzes auf dem Quellsystem. Wenn dieser Zeitstempel älter oder gleich dem abgespeicherten TimeStamp in CMI LehrerOffice ist, wird dieses Element und alle Unterelemente nicht verarbeitet. Dank diesem TimeStamp kann die Verarbeitungszeit stark optimiert werden.

Dieser Zeitstempel gilt nur für die Klassendefinition und nicht für die Schülerdaten.

Format: YYYY-MM-DDTHH:MM:SS

Beispiel: 2010-01-03T09:32:59

3.3.7 Elemente zu 'classDefinition'

name

zwingend notwendig

Allgemeingültige Klassenbezeichnung als Kürzel, wobei semesterabhängige Angaben und Namen von Lehrpersonen zu unterlassen sind.

Format: alphanumerisch

Beispiel: 2. Real b

grade

zwingend notwendig

Definition der entsprechenden Schulstufe.

Werte: VS = Kindergarten / Vorschulstufe und Basisstufe

PS = Primarstufe

SEK1 = Sekundarstufe I

SEK2 = Sekundarstufe II

Beispiel: SEK1

classForm zwingend notwendig

Art der Schulklasse.

Werte: **KG** = Kindergarten
BS = Basisstufe
ES = Einschulungsklasse
KK[n] = Kleinklasse A..D
O = Sonderklasse
P = Primarklasse
S[n] = Sekundarklasse (Abteilung, Niveau als zweites Zeichen z.B. "SA")
R = Realklasse
W = Werkklasse

Beispiel: R

classFormCode empfohlen

BFS-Code der Klassenform

Format: **alphanumerisch**

Beispiel: 11

classFormName empfohlen

Vollständiger Name der Klassenform

Format: **alphanumerisch**

Beispiel: Kindergarten

classYear zwingend notwendig

Schulkasse resp. Jahr der entsprechenden Schulstufe. Bei einer 2. Sekundarklasse ist somit eine 2 zu liefern.

Format: **numerisch (1)**

Beispiel: 2

classAppendix optional

Bei Parallelklassen kann ein Zusatz wie 'a' übergeben werden, damit eine Unterscheidung möglich wird.

Format: **alphanumerisch**

Beispiel: b

unitKey optional

Grosse Schulgemeinden können in Schuleinheiten unterteilt werden, wobei jede Klasse einer Schuleinheit zugeordnet wird.

Eindeutige ID vom Quellsystem für die Schuleinheit. Die Schuleinheiten müssen vorab innerhalb von CMI Lehrer-Office definiert und mit demselben Schlüsselwert wie im Quellsystem eingestellt werden.

Format: **alphanumerisch**

Beispiel: OZ1

unitName optional

Name der Schuleinheit. Dient lediglich als Information für die Zuordnung der Schuleinheit zur ID, wird nicht übernommen.

Format: **alphanumerisch**

Beispiel: Sekundarschule

unitCity

optional

Schulort der Schuleinheit.

Format: `alphanumerisch`

Beispiel: `Thayngen`

bur="72765790"

optional

Zur Identifikation wird die BUR-Nummer SSP vom Amt für Statistik verwendet. Diese BUR-Nummer SSP ist ein eindeutiger Identifikator der Bildungsinstitute der Schweiz.

Format: `numerisch (20)`

Beispiel: `72765790`

<code><classData></code>	<i>Stammdaten der Klasse</i>
--------------------------------	------------------------------

lehreroffice/inst/semesters/semester/schoolclasses/schoolclass/classData

3.3.8 Element 'classData'

empfohlen

Im Gegensatz zum Element 'classDefinition' werden in diesem Element die semesterabhängigen Grunddaten der aktuellen Schulklasse überliefert.

Anzahl: 1

Beispiel:

```
<classData>
  ...
</classData>
```

3.3.9 Elemente zu 'classData'

schoolhouse

empfohlen

Bezeichnung des Schulhauses, welche je nach Kanton im Zeugnisdruck verwendet wird. Diese Information wird innerhalb CMI LehrerOffice semesterabhängig abgelegt.

Format: [alphanumerisch](#)

Beispiel: [Wisli](#)

schoolhouseKey

empfohlen

Eindeutige ID vom Quellsystem für das Schulhaus. Die Schulhäuser müssen vorab innerhalb von CMI LehrerOffice definiert und mit demselben Schlüsselwert wie im Quellsystem eingestellt werden.

Format: [alphanumerisch](#)

Beispiel: [WSI](#)

schoolhouseCity

optional

Schulort des Schulhauses.

Format: [alphanumerisch](#)

Beispiel: [Thayngen](#)

room

optional

Nummer des Schulzimmers / Klassenzimmers.

Format: [alphanumerisch](#)

Beispiel: [A21](#)

4 Daten einer Schulklasse mit Schülerdaten

Mit dieser Methode können die Schülerdaten von der Schulverwaltung übernommen werden. Diese Angaben werden schlussendlich im Modul 'Personalnoten' von CMI LehrerOffice abgelegt werden.

Daten einer Schulklasse

- Schülerdaten mit Kontaktadressen
- Klassen- und Fachlehrpersonenzuteilungen
- Niveau- und Wahlfachzuteilungen

SOAP-Methode: `LO_getSchoolclass`

Parameter: `int semesterYear, int semesterPart, String classKey`

Alternative SOAP-Methode: `LO_getSchoolclass2`

Parameter: `int semesterYear, int semesterPart, String classKey, String classID`

Dateiname: `"SJ" + Schuljahr + "-" + Semester + "_Klasse_" + Klassenkürzel + ".xml"`

Beispiel: `SJ2011-2_Klasse_P1a.xml`

Es ist wichtig, dass die Daten für die Lehrpersonen zuerst für den Import zur Verfügung gestellt werden, da die Lehrpersonendaten für die Zuteilung dieser Lehrpersonen nachfolgend von den Klassendaten vorhanden sein müssen!

4.1 SOAP – Anfrage für Daten einer Schulklasse mit Schülerdaten

<LO_getSchoolclass>

Methodenaufruf mit Parameter

soap:Envelope/soap:Body/LO_getSchoolclass

4.1.1 Element 'LO_getSchoolclass'

zwingend notwendig

Dieses Element ist der eigentliche Methodenaufruf für die Abfrage/Aktion der Klassendaten mit den dazugehörigen Schülern.

Jede Schulverwaltungslösung stellt diese SOAP-Methode als Web-Service zur Verfügung. Das erwartete Resultat als XML ist im Abschnitt 'Daten einer Schulklasse – XML-Struktur' dokumentiert.

Weitere SOAP-Methoden und deren Parameter finden Sie nachfolgend beschrieben.

Beispiel:

```
<soap:Body>
  <LO_getSchoolclass xmlns="http://www.lehrer-
office.ch/xmlns/soap/2010/exchange-v1">
    <semesterYear>2010</semesterYear>
    <semesterPart>1</semesterPart>
    <classKey>SHA-PS-1-b</classKey>
  </LO_getSchoolclass>
</soap:Body>
```

4.1.2 Attribute zum Element 'LO_getSchoolclass'

xmlns="http://www.lehreroffice.ch/xmlns/soap/2010/exchange-v1"

zwingend notwendig

Dieser Namespace muss als Standardnamespace, also ohne Namespace-Präfix angegeben werden.

Diese Versionsangabe hat keinen Zusammenhang mit der Programmversion von CMI LehrerOffice als Anwendung.

Format: URL

Beispiel: <http://www.lehreroffice.ch/xmlns/soap/2010/exchange-v1>

4.1.3 Elemente zu 'LO_getSchoolclass' als Parameter

`semesterYear` zwingend notwendig

Vierstellige Ganzzahl für das Schuljahr, wobei das Kalenderjahr des Schulanfangs geliefert wird. Für das Schuljahr 2010/11 wird somit die Zahl '2010' geliefert.

Format: `numerisch (4)`

Beispiel: `2010`

`semesterPart` zwingend notwendig

Einstellige Ganzzahl für die Semesterangabe.

Format: `1|2`

Beispiel: `1`

`classKey` zwingend notwendig

Eindeutiger Schlüssel für die Identifikation der Schulklasse.

Eine detaillierte Beschreibung zu diesem Schlüsselwert finden Sie im Abschnitt "Attribute zum Element 'schoolclass'" auf Seite 51.

Format: `alphanumerisch`

Beispiel: `SHA-PS-1-b`

`classID` optional

Eindeutige ID bei einem Kurssystem mit laufender, schuljahresabhängiger ID (nur für EcoOpen und SclarisWeb). Dieser Parameter wird nur geliefert, wenn in CMI LehrerOffice eine laufende ID erfasst wurde.

Weitere Informationen sind im Abschnitt "Attribute zum Element 'schoolclass'" auf Seite 5 zu finden.

Format: `alphanumerisch`

Beispiel: `1004`

4.2 Daten einer Schulklasse – XML-Struktur

XML-Tag	Dateninhalt
<lehreroffice>	Dateninhalt
<inst>	Institution / Schulgemeinde
<source>	Angaben zum Quellsystem
</source>	
<semesters>	Auflistung der Semester
<semester>	Semesterdaten
<schoolclasses>	Auflistung der Schulklassen
<schoolclass>	Schulklasse
<classDefinition>	Klassendefinition
</classDefinition>	
<classData>	Stammdaten der Klasse
</classData>	
<classTeachers>	Auflistung aller Klassenlehrpersonen
<classTeacher>	Klassenlehrperson
</classTeacher>	
</classTeachers>	
<subjects>	Auflistung aller Schulfächer
<subject>	Zuteilung der Fachlehrpersonen
<subjectTeachers>	Auflistung aller Fachlehrpersonen
<subjectTeacher>	Fachlehrperson
</subjectTeacher>	
</subjectTeachers>	
</subject>	
</subjects>	
<students>	Auflistung aller Schüler/innen
<student>	Schüler/in
<studentData>	Stammdaten des Schülers
<homeAddress>	Wohnanschrift des Schülers
</homeAddress>	
<freeFields>	Freifelder mit Zusatzinformationen
</freeFields>	
<entrance>	Daten zum Schuleintritt
</entrance>	
<leaving>	Daten zum Schulaustritt
</leaving>	
</studentData>	
</student>	
</students>	
</schoolclass>	
</schoolclasses>	
</semester>	
</semesters>	
</inst>	
</lehreroffice>	

Fortsetzung auf nachfolgender Seite...

Fortsetzung von vorangehender Seite...

<contacts>	Auflistung der Eltern und Kontaktpersonen
<contact>	Angaben zum Elternteil resp. Kontaktperson
<homeAddress>	Private Wohnanschrift
</homeAddress>	
<workAddress>	Geschäftsanschrift
</workAddress>	
</contact>	
<contacts>	
<history>	Schülergeschichte mit alten Wohnadressen
<addressItem>	Frühere Wohnadresse
</addressItem>	
</history>	
<schoolData>	Schulform (bei integrierten Schülern)
<individualFostering>	Fördermassnahmen
</individualFostering>	
</schoolData>	
<subjects>	Auflistung der Schulfächer
<subject>	Fach- und Niveauzuteilung des Schülers
</subject>	
</subjects>	
</student>	
</students>	
</schoolclass>	
</schoolclasses>	
</semester>	
</semesters>	
</inst>	
</lehreroffice>	

4.3 Daten einer Schulklasse – Elemente



Obige Elemente und dessen Attribute und Unterelemente sind im vorangehenden Kapitel ' Daten für Lehrpersonen – Elemente' detailliert dokumentiert. Nachfolgend sind nun die Elemente innerhalb vom Element '<semester>' beschrieben.



lehreroffice/inst/semesters/semester/schoolclasses

4.3.1 Element 'schoolclasses'

zwingend notwendig

Dieses Element fungiert als Liste mit Unterelementen vom Typ '<schoolclass>'. In der aktuellen Version wird nur eine einzelne Schulklasse unterstützt. In künftigen Versionen sind unter Umständen mehrere Schulklassen innerhalb einer XML-Datei möglich.

Anzahl: 1

Beispiel:

```
<schoolclasses>
  <schoolclass>
    ...
  </schoolclass>
</schoolclasses>
```

<code><schoolclass></code>	<i>Schulklasse</i>
----------------------------------	--------------------

lehreroffice/inst/semesters/semester/schoolclasses/schoolclass

4.3.2 Element 'schoolclass'

zwingend notwendig

Datenelement einer Schulklasse. Es wird empfohlen pro Klasse eine XML-Datei zu exportieren. Dies vereinfacht die allfällige Fehlersuche das wiederholte Einlesen von Klassendaten.

Anzahl: 1..n

Beispiel:

```
<schoolclass key="SHA-PS-1-b" id="9" active="1" modDateTime="2010-01-03T09:32:59">
  ...
</schoolclass>
```

4.3.3 Attribute zum Element 'schoolclass'

key="SHA-PS-1-b"

zwingend notwendig

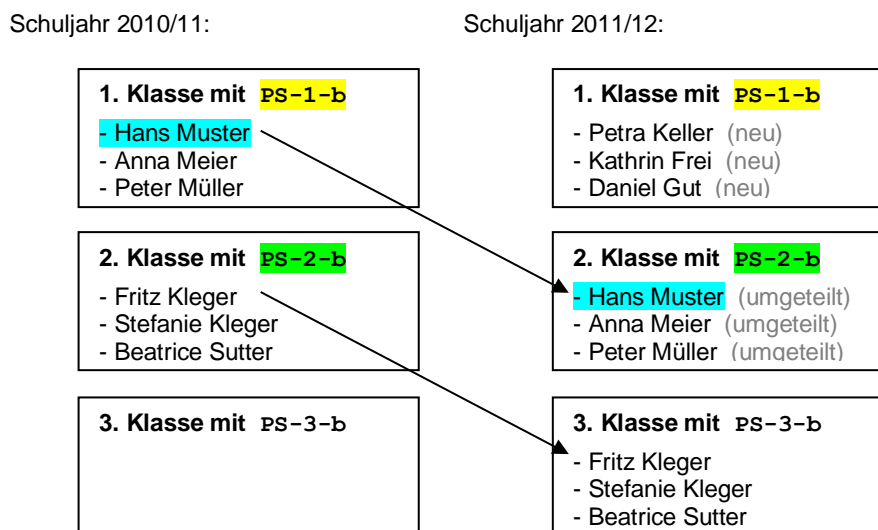
Eindeutige und statische ID vom Quellsystem für die Schulklasse, welche über alle Schuljahre z.B. für die 1. Primar-klasse unverändert bleibt.

Format: **alphanumerisch**

Beispiel: **SHA-PS-1-b**

In CMI LehrerOffice werden die Klassengefäße einmalig definiert, wobei dazu für jede Klasse die dazugehörigen Schulfächer, Lernziele und allenfalls Kompetenzraster hinterlegt werden. Die einzelnen Schüler und die entsprechenden Klassenlehrer werden am Ende eines Schuljahres nun von der aktuellen Klassendefinition zur nächsten weitergeschoben. Vergleichbar mit einem Klassenzimmer, das immer wieder mit neuen Schülern und Lehrpersonen belegt wird.

Die Klassendefinitionen mit ihren internen ID's bleiben somit statisch und für jedes Schuljahr grundsätzlich identisch. Siehe nachfolgendes Beispiel:



Diese ID kann selbstverständlich auch aus verschiedenen Feldern zusammengestellt werden, wobei eine eindeutige ID sichergestellt werden muss. Ein möglicher Ansatz ist:

Schuleinheit + "-" + Stufe + "-" + Klasse (Integer) + "-" + Gruppe

Beispiele: SHA-PS-2-b (Schulhaus Amden, Primarschule, 2. Klasse b)
OZ1-REAL-1-a (Oberstufenzentrum 1, Real, 1. Klasse a)

Der Schlüsselwert muss in CMI LehrerOffice gemäss Klassenliste der Schulverwaltungslösung im Modul 'Klassen' im Feld 'Schlüssel' einmalig erfasst werden, wobei die Gross- und Kleinschreibung nicht berücksichtigt wird. Ein direkter Import mit automatischer Erstellung von neuen Schulklassen in CMI LehrerOffice ist vorerst nicht geplant, da die Fachdefinitionen und Kompetenzraster entsprechend zugewiesen werden müssen.

Wenn die Schulklasse mit dem identischen Key nicht in CMI LehrerOffice gefunden wird, dann werden alle nachfolgenden Schülerdaten dieser Schulklasse nicht importiert.

id optional

Bei Schulverwaltungslösungen (wie Scolaris), welche für jede Schulklasse eine neue ID zu Beginn der Schulzeit generieren, kann die laufende ID in CMI LehrerOffice auch als zweites Suchkriterium genutzt werden.

Dies bedeutet, dass z.B. bei einer 1. Primarklasse die ID mit "9" in CMI LehrerOffice als laufenden Fremdschlüssel bei der entsprechenden 1. Klasse erfasst werden kann.

Schulverwaltungslösungen (wie iCampus), welche dasselbe System mit den fixen Schlüsselwerten wie CMI LehrerOffice anwendet, darf dieses Attribut nicht nutzen.

Format: alphanumerisch

Beispiel: 9

modDateTime optional

Modifikationsdatum im XSD-Format. Letzte Mutation des Datensatzes auf dem Quellsystem. Wenn dieser Zeitstempel älter oder gleich dem abgespeicherten TimeStamp in CMI LehrerOffice ist, wird dieses Element und alle Unterelemente nicht verarbeitet. Dank diesem TimeStamp kann die Verarbeitungszeit stark optimiert werden.

Auch bei einer Änderung der dazugehörigen Klassen- oder Fachlehrpersonen, der untergeordneten Schülerdaten oder Fachzuteilungen muss ein geänderter Zeitstempel geliefert werden!

Format: YYYY-MM-DDTHH:MM:SS

Beispiel: 2010-01-03T09:32:59

active optional

Schulklasse ist in diesem Semester aktiv (1) gekennzeichnet oder nicht (0).

Format: numerisch 0|1 (1)

Beispiel: 1

<code><classDefinition></code>	<i>Klassendefinition</i>
--------------------------------------	--------------------------

lehreroffice/inst/semesters/semester/schoolclasses/schoolclass/classDefinition

4.3.4 Element 'classDefinition'

(nicht implementiert) optional

Damit künftig Klassengefässe automatisch in CMI LehrerOffice erstellt werden können, kann die Definition der Klasse mit diesen Angaben einmalig erstellt werden, wenn die Klasse gemäss Schlüssel nicht gefunden werden konnte. Diese Definitionen sind im Gegensatz zu 'classData' nicht semesterabhängig.

Anzahl: 1

Beispiel:

```
<classDefinition>
  ...
</classDefinition>
```

4.3.5 Attribute zu 'classDefinition'

modDateTime

optional

Modifikationsdatum im XSD-Format. Letzte Mutation des Datensatzes auf dem Quellsystem. Wenn dieser Zeitstempel älter oder gleich dem abgespeicherten TimeStamp in CMI LehrerOffice ist, wird dieses Element und alle Unter-elemente nicht verarbeitet. Dank diesem TimeStamp kann die Verarbeitungszeit stark optimiert werden.

Dieser Zeitstempel gilt nur für die Klassendefinition und nicht für die Schülerdaten.

Format: YYYY-MM-DDTHH:MM:SS

Beispiel: 2010-01-03T09:32:59

4.3.6 Elemente zu 'classDefinition'

name

optional

Allgemeingültige Klassenbezeichnung als Kürzel, wobei semesterabhängige Angaben und Namen von Lehrpersonen zu unterlassen sind.

Format: alphanumerisch

Beispiel: 2. Real b

grade

zwingend notwendig

Definition der entsprechenden Schulstufe.

Werte: VS = Kindergarten / Vorschulstufe und Basisstufe

PS = Primarstufe

SEK1 = Sekundarstufe I

SEK2 = Sekundarstufe II

Beispiel: SEK1

classForm zwingend notwendig

Art der Schulklasse.

Werte:

KG	=	Kindergarten
BS	=	Basisstufe
ES	=	Einschulungsklasse
KK[n]	=	Kleinklasse A..D
O	=	Sonderklasse
P	=	Primarklasse
S[n]	=	Sekundarklasse (Abteilung, Niveau als zweites Zeichen z.B. "SA")
R	=	Realklasse
W	=	Werkklasse

Beispiel: R

classFormCode empfohlen

BFS-Code der Klassenform

Format: `alphanumerisch`

Beispiel: 11

classFormName empfohlen

Vollständiger Name der Klassenform

Format: `alphanumerisch`

Beispiel: Kindergarten

classYear zwingend notwendig

Schulklasse resp. Jahr der entsprechenden Schulstufe. Bei einer 2. Sekundarklasse ist somit eine 2 zu liefern.

Format: `numerisch (1)`

Beispiel: 2

classAppendix optional

Bei Parallelklassen kann ein Zusatz wie 'a' übergeben werden, damit eine Unterscheidung möglich wird.

Format: `alphanumerisch`

Beispiel: b

unitKey optional

Grosse Schulgemeinden können in Schuleinheiten unterteilt werden, wobei jede Klasse einer Schuleinheit zugeordnet wird.

Eindeutige ID vom Quellsystem für die Schuleinheit. Die Schuleinheiten müssen vorab innerhalb von CMI Lehrer-Office definiert und mit demselben Schlüsselwert wie im Quellsystem eingestellt werden.

Format: `alphanumerisch`

Beispiel: OZ1

unitName empfohlen

Name der Schuleinheit. Dient lediglich als Information für die Zuordnung der Schuleinheit zur ID, wird nicht übernommen.

Format: `alphanumerisch`

Beispiel: Sekundarschule

bur="72765790"

optional

Zur Identifikation wird die BUR-Nummer SSP vom Amt für Statistik verwendet. Diese BUR-Nummer SSP ist ein eindeutiger Identifikator der Bildungsinstitute der Schweiz.

Format: `numerisch (20)`

Beispiel: `72765790`

<code><classData></code>	<i>Stammdaten der Klasse</i>
--------------------------------	------------------------------

lehreroffice/inst/semesters/semester/schoolclasses/schoolclass/classData

4.3.7 Element 'classData'

empfohlen

Im Gegensatz zum Element 'classDefinition' werden in diesem Element die semesterabhängigen Grunddaten der aktuellen Schulklasse überliefert.

Anzahl: 1

Beispiel:

```
<classData>
  ...
</classData>
```

4.3.8 Elemente zu 'classData'

schoolhouse

empfohlen

Bezeichnung des Schulhauses, welche je nach Kanton im Zeugnisdruck verwendet wird. Diese Information wird innerhalb CMI LehrerOffice semesterabhängig abgelegt.

Format: `alphanumerisch`

Beispiel: `Wisli`

schoolhouseKey

empfohlen

Eindeutige ID vom Quellsystem für das Schulhaus. Die Schulhäuser müssen vorab innerhalb von CMI LehrerOffice definiert und mit demselben Schlüsselwert wie im Quellsystem eingestellt werden.

Format: `alphanumerisch`

Beispiel: `WSI`

room

optional

Nummer des Schulzimmers / Klassenzimmers.

Format: `alphanumerisch`

Beispiel: `A21`

<code><classTeachers></code>	<i>Auflistung aller Klassenlehrpersonen</i>
------------------------------------	---

lehreroffice/inst/semesters/semester/schoolclasses/schoolclass/classTeachers

4.3.9 Element 'classTeachers'

empfohlen

Dieses Element fungiert als Liste mit den entsprechenden Adresselementen vom Typ '<classTeacher>'. Diese Liste umfasst alle Klassenlehrpersonen und Heilpädagogen dieser Klasse. Die Fachlehrpersonen werden separat übermittelt (siehe Abschnitt 'subjects').

Die Lehrpersonen müssen vorab als Adressen innerhalb von CMI LehrerOffice vorhanden sein. Aus diesem Grund werden die Lehrpersonenadressen zuerst importiert, damit die aktuellen Adressdaten der Lehrpersonen zur Verfügung stehen.

Anzahl: 1

Beispiel:

```
<classTeachers complete="1">  
  <classTeacher ahv="756.0978.6543.21" key="1001" function="KL" />  
  <classTeacher key="1323" function="WKL" />  
</classTeachers>
```

4.3.10 Attribute zum Element 'classTeachers'

`complete="1"`

zwingend notwendig

Wenn der Code "1" (Standard) übergeben wird, dann ist die nachfolgende Liste mit den Klassenlehrpersonen vollständig. Dies bedeutet, dass alle Klassenlehrpersonen enthalten sind. Überzählige Lehrpersonen werden hiermit automatisch innerhalb CMI LehrerOffice von dieser Klasse entfernt.

Mit dem Code "0" werden die fehlenden Lehrpersonen hinzugefügt. Diese Option ist für das Nachliefern von einzelnen Lehrpersonen sinnvoll (Push-Variante).

- 0 = Liste mit Lehrpersonen ist nicht vollständig
- 1 = Liste mit Lehrpersonen ist komplett (Standard)

Format: `numerisch 0|1 (1)`

Beispiel: 1

<code><classTeacher></code>	<i>Klassenlehrperson</i>
-----------------------------------	--------------------------

lehreroffice/inst/semesters/semester/schoolclasses/schoolclass/classTeachers/
classTeacher

4.3.11 Element 'classTeacher'

empfohlen

Datenelement für das Zuordnen einer Klassenlehrperson. Dieses Element enthält keine weiteren Elemente, sondern dient lediglich für die Zuteilung der Lehrpersonen, welche im Adressstamm bereits enthalten sein müssen.

Anzahl: 0..5

Beispiel:

```
<classTeacher ahv="756.0978.6543.21" key="1001" function="KL" />
```

4.3.12 Attribute zum Element 'classTeacher'

ahv="756.0978.6543.21"

empfohlen

AHV-Nummer dient als Identifikator mit erster Priorität. Die Datenschnittstelle von CMI LehrerOffice sucht zuerst nach einem Adressen-Datensatz mit entsprechender AHV-Nummer. Wenn erfolgreich, wird diese Adresse als Klassenlehrperson zugeteilt. Sollte der Datensatz nicht gefunden werden, wird in zweiter Priorität nach dem Attribut 'key' gesucht (siehe nachfolgende Beschreibung).

Format: alphanumerisch (16)

Beispiel: 756.0978.6543.21

key="1001"

zwingend notwendig

Eindeutige ID vom Quellsystem für die entsprechende Adresse der Lehrperson.

Wenn die Schulverwaltung noch keine gültigen AHV-Nummern besitzt kann in zweiter Priorität mit diesem Schlüsselwert abgeglichen werden. Sobald eine AHV-Nummer vorhanden ist, wird zuerst nach dieser gesucht.

Format: alphanumerisch

Beispiel: 1001

function="KL"

zwingend notwendig

Funktion resp. Rolle dieser Lehrperson

Werte:	KL	=	Klassenlehrperson als verantwortliche Hauptlehrperson (max. 6 LP)
	WKL	=	Weitere Klassenlehrperson bei Jobsharing (max. 6 LP)
	STV	=	Stellvertretung, Vikariat (1 LP)
	HP	=	Schulischer Heilpädagoge/in (2 LP)
	SSA	=	Schulsozialarbeiter (1 P)
	WFP	=	Förderlehrperson (3 LP)

Beispiel: KL

`<subjects>`

Auflistung aller Schulfächer

lehreroffice/inst/semesters/semester/schoolclasses/schoolclass/subjects

4.3.13 Element 'subjects'

empfohlen

Dieses Element fungiert als Liste mit den entsprechenden Elementen vom Typ '<subject>'. Diese Liste umfasst die Schulfächer einer Klasse, damit die Fachlehrpersonen zugeteilt werden können.

Anzahl: 1

Beispiel:

```
<subjects>
  ...
</subjects>
```

`<subject>`

Zuteilung der Fachlehrpersonen

lehreroffice/inst/semesters/schoolclasses/schoolclass/subjects/subject

4.3.14 Element 'subject'

wenn Daten vorhanden

Datenelement eines einzelnen Schulfaches, das in der entsprechenden Schulklasse unterrichtet wird.

Anzahl: 0..n

Beispiel: `<subject key="D" caption="Deutsch">`
...
`</subject>`

4.3.15 Attribute zum Element 'subject'

`key="D"`

zwingend notwendig

Eindeutige und statische ID vom Quellsystem für das Schulfach, welche über alle Schuljahre unverändert bleibt.

Format: `alphanumerisch`

Beispiel: `D`

In CMI LehrerOffice werden die Schulfächer einmalig definiert und einmalig mit dem dazugehörigen Schlüssel der Schulverwaltung ergänzt.

Bei den Fachdefinitionen darf der Wert im Feld 'Schlüssel' von CMI LehrerOffice nicht verändert werden, da sonst der Zeugnisdruck nicht mehr korrekt funktionieren wird! Für den Abgleich wird ein zweiter, separater Schlüsselwert von CMI LehrerOffice zur Verfügung gestellt, der frei an die Schulverwaltung angepasst werden darf.

`caption="Deutsch"`

empfohlen

Bezeichnung des Schulfachs, damit der gelieferte Fachschlüssel in CMI LehrerOffice am richtigen Ort eingetragen werden kann.

Format: `alphanumerisch`

Beispiel: `Deutsch`

<code><subjectTeachers></code>	<i>Auflistung aller Fachlehrpersonen</i>
--------------------------------------	--

lehreroffice/inst/semesters/semester/schoolclasses/schoolclass/subjects/subject/subjectTeachers

4.3.16 Element 'subjectTeachers'

empfohlen

Dieses Element fungiert als Liste mit den entsprechenden Adresselementen vom Typ '<subjectTeacher>'. Diese Liste umfasst alle Fachlehrpersonen dieses Schulfaches. Die Klassenlehrpersonen werden separat übermittelt (siehe vorangehenden Abschnitt).

Die Lehrpersonen müssen vorab als Adressen innerhalb von CMI LehrerOffice vorhanden sein. Aus diesem Grund werden die Lehrpersonenadressen zuerst importiert, damit die aktuellen Adressdaten der Lehrpersonen zur Verfügung stehen.

Anzahl: 1

Beispiel:

```
<subjectTeachers complete="1">  
  <subjectTeacher ahv="756.0978.6543.21" key="1001" function="KL" />  
  <subjectTeacher key="1323" function="WKL" />  
</subjectTeachers>
```

4.3.17 Attribute zum Element 'subjectTeachers'

`complete="1"`

zwingend notwendig

Wenn der Code "1" (Standard) übergeben wird, dann ist die nachfolgende Liste mit den Fachlehrpersonen vollständig. Dies bedeutet, dass alle Fachlehrpersonen enthalten sind. Überzählige Lehrpersonen werden hiermit automatisch innerhalb CMI LehrerOffice von diesem Schulfach entfernt.

Mit dem Code "0" werden die fehlenden Lehrpersonen hinzugefügt. Diese Option ist für das Nachliefern von einzelnen Lehrpersonen sinnvoll (Push-Variante).

0 = Liste mit Lehrpersonen ist nicht vollständig

1 = Liste mit Lehrpersonen ist komplett (Standard)

Format: `numerisch 0|1 (1)`

Beispiel: 1

<code><subjectTeacher></code>	<i>Fachlehrperson</i>
-------------------------------------	-----------------------

lehreroffice/inst/semesters/semester/schoolclasses/schoolclass/subjects/subject/subjectTeachers/subjectTeacher

4.3.18 Element 'subjectTeacher'

empfohlen

Datenelement für das Zuordnen einer Fachlehrperson. Dieses Element enthält keine weiteren Elemente, sondern dient lediglich für die Zuteilung der Lehrpersonen, welche im Adressstamm bereits enthalten sein müssen.

Anzahl: 0..3

Beispiel:

```
<subjectTeacher ahv="756.0978.6543.21" key="1001" function="FL" />
```

4.3.19 Attribute zum Element 'subjectTeacher'

ahv="756.0978.6543.21"

empfohlen

AHV-Nummer dient als Identifikator mit erster Priorität. Die Datenschnittstelle von CMI LehrerOffice sucht zuerst nach einem Adressen-Datensatz mit entsprechender AHV-Nummer. Wenn erfolgreich, wird diese Adresse als Fachlehrperson zugeteilt. Sollte der Datensatz nicht gefunden werden, wird in zweiter Priorität nach dem Attribut 'key' gesucht (siehe nachfolgende Beschreibung).

Format: [alphanumerisch \(16\)](#)

Beispiel: [756.0978.6543.21](#)

key="1001"

zwingend notwendig

Eindeutige ID vom Quellsystem für die entsprechende Adresse der Lehrperson.

Wenn die Schulverwaltung noch keine gültigen AHV-Nummern besitzt kann in zweiter Priorität mit diesem Schlüsselwert abgeglichen werden. Sobald eine AHV-Nummer vorhanden ist, wird zuerst nach dieser gesucht.

Format: [alphanumerisch](#)

Beispiel: [1001](#)

function="FL"

zwingend notwendig

Funktion resp. Rolle dieser Lehrperson (weitere Rollen werden bei Bedarf in einer künftigen Version hinzugefügt.)

Werte: [FL](#) = Fachlehrperson

Beispiel: [FL](#)

<code><students></code>	<i>Auflistung aller Schüler/innen</i>
-------------------------------	---------------------------------------

lehreroffice/inst/semester/semesters/schoolclasses/schoolclass/students

4.3.20 Element 'students'

zwingend notwendig

Dieses Element fungiert als Liste mit den entsprechenden Elementen vom Typ '<student>'. Diese Liste umfasst die Schüler/innen und dessen Niveau- und Wahlfachzuteilungen der entsprechenden Schulklasse.

Anzahl: 1

Beispiel:

```
<students complete="1">  
  ...  
</students>
```

4.3.21 Attribute zum Element 'students'

`complete="1"`

zwingend notwendig

Wenn der Code "1" (Standard) übergeben wird, dann ist die nachfolgende Schülerliste vollständig. Dies bedeutet, dass alle Schüler von entsprechender Schulklasse enthalten sind. Überzählige Schüler innerhalb von CMI Lehrer-Office werden hiermit automatisch archiviert und aus der Klasse entfernt.

Mit dem Code "0" werden die fehlenden Schüler hinzugefügt. Diese Option ist für das Nachliefern von einzelnen Schülern sinnvoll (Push-Variante).

- 0 = Schülerliste ist nicht vollständig
- 1 = Schülerliste ist komplett (Standard)

Format: `numerisch 0|1 (1)`

Beispiel: 1

<code><student></code>	<i>Schüler/in</i>
------------------------------	-------------------

lehreroffice/inst/semesters/semester/schoolclasses/schoolclass/students/student

4.3.22 Element 'student'

wenn Daten vorhanden

Datenelement für die Schülerdaten sowie Niveau- und Wahlfachzuteilungen eines einzelnen Schülers.

Anzahl: 0..n

Beispiel:

```
<student ahv="456.1941.1243.92" key="9303"
      modDateTime="2010-01-03T09:32:59">
  ...
</student>
```

4.3.23 Attribute zum Element 'student'

ahv="456.1941.1243.92"

empfohlen

AHV-Nummer des Schülers dient als Identifikator mit erster Priorität. Die Datenschnittstelle von CMI LehrerOffice sucht zuerst nach einem gültigen Schüler-Datensatz mit entsprechender AHV-Nummer. Sollte der Datensatz nicht gefunden werden, wird in zweiter Priorität nach dem Attribut 'key' gesucht (siehe nachfolgende Beschreibung).

Format: [alphanumerisch \(16\)](#)

Beispiel: 456.1941.1243.92

key="9303"

zwingend notwendig

Eindeutige ID vom Quellsystem für den Schüler.

Wenn die Schulverwaltung noch keine gültigen AHV-Nummern besitzt kann in zweiter Priorität mit diesem Schlüsselwert abgeglichen werden. Sobald eine AHV-Nummer vorhanden ist, wird zuerst nach dieser gesucht.

Format: [alphanumerisch](#)

Beispiel: 9303

id

optional

Eindeutige ID für den Schüler. Diese kann für weitere Systeme wie zB. NEST genutzt werden. Diese ID wird im CMI LehrerOffice als Fremdschlüssel gespeichert und als Einwohnerkontrolle im Modul „Personaldaten“ angezeigt, aber nicht für die Identifikation des Schülers beim Datenimport genutzt.

Format: [alphanumerisch](#)

Beispiel: 98402

modDateTime

empfohlen

Modifikationsdatum im XSD-Format. Letzte Mutation des Datensatzes auf dem Quellsystem.

Auch bei einer Änderung der Zuordnung von Wahlfächern oder einer Kontaktadresse muss ein geänderter Zeitstempel geliefert werden!

Format: [YYYY-MM-DDTHH:MM:SS](#)

Beispiel: 2010-01-03T09:32:59

action=""

optional

Bearbeitungsaktion für Push-Variante, wenn die Mutationen laufend geliefert werden, denn hiermit müssen Löschaaktionen bewusst geliefert werden.

Standardmässig wird keine Aktion angegeben, da die überzähligen Schüler einer Klasse von CMI LehrerOffice ermittelt und archiviert werden, sofern das Attribut 'complete' im Element 'students' mit dem Wert "1" gesetzt wurde.

Werte: `del` = Datensatz wird gelöscht (betrifft alle Semester)
`hide` = Datensatz wird ausgeblendet (betrifft alle Semester)

<studentData>

Stammdaten des Schülers

lehreroffice/inst/semesters/semester/schoolclasses/schoolclass/students/student/studentData

4.3.24 Element 'studentData'

zwingend notwendig

In diesem Element werden die semesterunabhängigen Grunddaten des Schülers abgelegt. Die Kontakt- und Adressangaben der Erziehungsberechtigten werden im separaten und nachfolgenden Element 'contacts' geliefert.

Anzahl: 1

Beispiel:

```
<studentData>
  ...
</studentData>
```

4.3.25 Elemente zu 'studentData'

prename

zwingend notwendig

Offizieller Vorname des Schülers. Diese Angabe wird für den Zeugnisdruck verwendet.

Format: `alphanumerisch`

Beispiel: `Anna Bertha`

name

zwingend notwendig

Nachname des Schülers. Diese Angabe wird für den Zeugnisdruck verwendet.

Format: `alphanumerisch`

Beispiel: `Meier`

gender

zwingend notwendig

Geschlecht des Schülers. Diese Angabe wird für den Zeugnisdruck und für Statistiken verwendet.

Format: `alphanumerisch (1)`

Werte: `f` = weiblich (feminin)

`m` = männlich (masulin)

Beispiel: `f`

birthdate

zwingend notwendig

Geburtsdatum des Schülers. Diese Angabe wird für den Zeugnisdruck verwendet.

Format: `YYYY-MM-DD`

Beispiel: `1973-03-22`

nationality empfohlen

Nationalität des Schülers, wobei die Landesangabe ausgeschrieben erwartet wird.

Format: [alphanumerisch](#)

Beispiel: [Deutschland](#)

hometownCity empfohlen

Heimatort des Schülers. Bei ausländischen Schülern ohne Heimatort wird dieses Element leer oder nicht geliefert.

Format: [alphanumerisch](#)

Beispiel: [Erlinsbach](#)

hometownCanton empfohlen

Kanton des Heimatorts, wobei der Kürzel des Kantons mit maximal 2 Zeichen erlaubt ist. Bei ausländischen Schülern ohne Heimatort wird dieses Element leer oder nicht geliefert.

Format: [alphanumerisch \(2\)](#)

Beispiel: [AG](#)

confession empfohlen

Konfession des Schülers, wobei die Angabe als Adjektiv ausgeschrieben erwartet wird.

Format: [alphanumerisch](#)

Beispiel: [reformiert](#)

nativeLanguage empfohlen

Muttersprache des Schülers, wobei die Angabe als Nomen ausgeschrieben erwartet wird.

Format: [alphanumerisch](#)

Beispiel: [Spanisch](#)

colloquialLanguage optional

Umgangssprache des Schülers.

Format: [alphanumerisch](#)

Beispiel: [Deutsch](#)

aliasPrenome optional

Rufname des Schülers (Vorname), welcher in allen Listen mit der Ausnahme von offiziellen Dokumenten aufgeführt wird. Schüler von fremden Nationen besitzen oft mehrere, lange Vornamen, welche im alltäglichen Zusammenleben nicht verwendet werden. In der Regel ist ein Leerstring enthalten.

Format: [alphanumerisch](#)

Beispiel: [Anna](#)

aliasName optional

Deckname des Schülers resp. abgekürzter Nachname, welcher in allen Listen mit der Ausnahme von offiziellen Dokumenten aufgeführt wird. Es gibt spezielle Fälle, wo gefährdete Personen so geschützt werden. In der Regel ist ein Leerstring enthalten.

Format: [alphanumerisch](#)

Beispiel: [Bond](#)

emergencyNotice optional

Notfallhinweis, wie bei Allergien oder Krankheiten.

Format: [alphanumerisch](#)

Beispiel: [Allergie auf Bienenstiche](#)

mediaRepresentation optional

Einwilligung der Erziehungsberechtigten betreffend Mediendarstellung, wie das Veröffentlichen von Fotos oder Videos.

Format: 0|1|2

Werte:	0	=	Undefiniert, keine Einwilligung eingeholt (Defaultwert)
	1	=	Einwilligung für Veröffentlichung von Fotos und Videos vorhanden
	2	=	Keine Veröffentlichung von Fotos und Videos erwünscht

dentistKey optional

Zuständiger Zahnarzt des Schülers. Verknüpfung über den Schlüsselwert zu einer bestehenden Adresse eines Zahnarztes.

Beispiel: 98284

doctorKey optional

Zuständiger Hausarzt des Schülers. Verknüpfung über den Schlüsselwert zu einer bestehenden Adresse eines Arztes.

Format: alphanumerisch

Beispiel: 97001

<code><homeAddress></code>	<i>Wohnanschrift des Schülers</i>
----------------------------------	-----------------------------------

lehreroffice/inst/semesters/semester/schoolclasses/schoolclass/students/student/studentData/homeAddress

4.3.26 Element 'homeAddress'

empfohlen

Wohnadresse des Schülers. Diese Adressangaben sind unter Umständen identisch und redundant mit den Adressangaben der Kontaktperson (Mutter und Vater).

Anzahl: 1

Beispiel:

```
<homeAddress>
  <street>Unterdorf</street>
  <zip>8240</zip>
  <city>Thayngen</city>
  ...
</homeAddress>
```

4.3.27 Elemente zu 'homeAddress'

addressAddition

optional

Zusätzliche Adressangabe mit c/o-Angabe, Flurname oder Liegenschaftsname (für die Kantone Thurgau und Appenzell Innerrhoden).

Format: alphanumerisch

Beispiel: `Unteri Flur`

street

empfohlen

Strassenangabe ohne Hausnummer. Es wird empfohlen die Strasse ausgeschrieben (nicht 'Hauptstr.' sondern mit 'Hauptstrasse' zu liefern).

Wenn in der Schulverwaltung die Angaben nicht getrennt vorliegen kann notfalls auch die Strassenangabe mit Hausnummer geliefert werden. Die Strassenangabe wird schlussendlich aus den 3 Teilen 'street', 'streetNumber' und 'streetNumberAddition' zusammengesetzt.

Format: alphanumerisch

Beispiel: `Unterdorf`

streetNumber

empfohlen

Hausnummer der Strassenangabe.

Format: numerisch (4)

Beispiel: `2`

streetNumberAddition

optional

Zusatzangabe resp. Hausteil zur Hausnummer.

Format: alphanumerisch

Beispiel: `b`

poBox

optional

Postfachangabe, sofern vorhanden.

Format: alphanumerisch

Beispiel: `Postfach 776`

zip empfohlen

Postleitzahl. Es sind maximal 5 Stellen erlaubt (Deutschland).

Format: `numerisch (5)`

Beispiel: `8240`

city empfohlen

Ortsangabe des Wohnorts.

Format: `alphanumerisch`

Beispiel: `Bibern`

municipality empfohlen

Zugehörige politische Gemeinde

Format: `alphanumerisch`

Beispiel: `Thayngen`

canton optional

Wohnkanton.

Format: `alphanumerisch (2)`

Beispiel: `SH`

country optional

Angabe zum Land.

Format: `alphanumerisch`

Beispiel: `Schweiz`

transport optional

Angaben zum Verkehrsmittel für den Schulweg.

Format: `alphanumerisch (64)`

Beispiel: `Fahrrad`

distance optional

Angaben zur Distanz des Schulwegs.

Format: `0|10|20|30`

Werte:	0	=	Undefiniert (Defaultwert)
	10	=	Kurzer Schulweg
	20	=	Mittlerer Schulweg
	30	=	Langer Schulweg

phone empfohlen

Erstkontakt resp. Haupt- Telefonnummer der Familie resp. Erziehungsberechtigten.

Format: `alphanumerisch`

Beispiel: `052 734 14 25`

mobile empfohlen

Mobilnummer vom Handy des Schülers (nicht der Eltern!) als beliebiger String.

Format: `alphanumerisch`

Beispiel: `078 344 15 42`

fax

veraltet

Private Faxnummer. Dieses veraltete Feld wird in künftigen Programmversionen von CMI LehrerOffice nicht mehr genutzt werden.

Format: `alphanumerisch`

Beispiel: `052 333 30 29`

email

empfohlen

Persönliche E-Mailadresse des Schülers, die eine Schuladresse sein kann.

Format: `alphanumerisch`

Beispiel: `schueler.anna@schule.com`

website

optional

Persönliche Webseite des Schülers als vollständige URL, die eine Schuladresse sein kann.

Format: `alphanumerisch`

Beispiel: `http://www.schule.com/schueler/anna`

`<freeFields>`

Freifelder mit Zusatzinformationen

`lehreroffice/inst/semesters/semester/schoolclasses/schoolclass/students/student/studentData/freeFields`

4.3.28 Element 'freeFields'

optional

In CMI LehrerOffice können bei den Schülerstammdaten 3 Freifelder von der Schule frei definiert und so beliebige Zusatzangaben erfasst werden. Unter Umständen wird die Zahl der Freifelder in CMI LehrerOffice ausgebaut werden, deshalb kann bei der XML-Schnittstelle bereits eine Liste mit unbeschränkt vielen Feldern geliefert werden:

Anzahl: `1`

Beispiel:

```
<freeFields>
  <freeField key="1">anna.meier</freeField>
  <freeField key="2">1234</freeField>
  ...
</freeFields>
```

4.3.29 Elemente zu 'freeFields'

freeField

optional

Jedes einzelne Freifeld wird als reines Textfeld geliefert und so im entsprechenden Freifeld in CMI LehrerOffice abgelegt. Die Zuordnung erfolgt über das Attribut "key". Mit key="2" wird zum Beispiel das zweite Freifeld definiert. Wenn der als Textwert ein Leerstring geliefert wird, dann wird die allenfalls vorhandene Eingabe in CMI LehrerOffice gelöscht resp. mit dem Leerwert überschrieben werden!

Format: `alphanumerisch`

Beispiel: `1234`

<entrance>

Daten zum Schuleintritt

lehreroffice/inst/semesters/semester/schoolclasses/schoolclass/students/student/studentData/entrance

Element 'entrance'

optional

Angaben zum Schuleintritt. Der Schuleintritt kann von der Einschulung oder nach einem Umzug ausgelöst werden.

4.3.30 Elemente zum Element 'entrance'

date

optional

Datum vom Schuleintritt. Der Schuleintritt kann von der Einschulung oder nach einem Umzug ausgelöst werden.

Format: YYYY-MM-DD

Beispiel: 2010-08-04

schoolCity

optional

Bei einem Umzug wird hiermit der vorangehende Schulort definiert.

Format: alphanumerisch

Beispiel: Kreuzlingen

schoolCanton

optional

Bei einem Umzug wird hiermit der Kanton des vorangehenden Schulorts definiert.

Format: alphanumerisch (2)

Beispiel: TG

schoolCountry

optional

Angabe zum Land.

Format: alphanumerisch

Beispiel: Schweiz

schoolClass

optional

Vorangehende Schulklasse.

Format: alphanumerisch

Beispiel: Schweiz

<code><passovers></code>	<i>Daten zum Übertritt resp. Stufenwechsel</i>
--------------------------------	--

lehreroffice/inst/semesters/semester/schoolclasses/schoolclass/students/student/studentData/passovers

Element 'passovers' optional

Angaben zum Übertritt resp. Wechsel in eine neue Schulstufe. Diese Angaben werden hauptsächlich für das Personalblatt des Kantons Zug oder Kanton Thurgau benötigt.

Anzahl: 1 bis n

Beispiel:

```
<studentData>
  ...
  <passovers>
    <passover toGrade="SEK1">
      <date>2009-08-06</date>
      <schoolCity>Dorfschulhaus, Kreuzlingen</schoolCity>
    </passover>
  </passovers>
  ...
</ studentData >
```

4.3.31 Attribute zum Element 'passover'

toGrade zwingend notwendig

Definition der Schulstufe, in die der Schüler/die Schülerin wechselt.

- Werte:
- FRWVS = freiwilliger Kindergarten
 - VS = Kindergarten / Vorschulstufe und Basisstufe
 - PS = Primarstufe
 - SEK1 = Sekundarstufe I
 - SEK2 = Sekundarstufe II

Beispiel: SEK1

4.3.32 Elemente zum Element 'passover'

date optional

Datum des Zeitpunktes, wann der Übertritt stattgefunden hat.

Format: YYYY-MM-DD

Beispiel: 2009-08-06

schoolCity optional

Name der Schulgemeinde. Allenfalls mit Angabe zum Schulhaus.

Format: alphanumerisch

Beispiel: Dorfschulhaus, Kreuzlingen

schoolCanton optional

Kantonsangabe der Schulgemeinde.

Format: alphanumerisch (2)

Beispiel: TG

schoolCountry

optional

Angabe zum Land.

Format: `alphanumerisch`

Beispiel: `Schweiz`

<leaving>

Daten zum Schulaustritt

lehreroffice/inst/semesters/semester/schoolclasses/schoolclass/students/student/studentData/leaving

Element 'leaving'

optional

Angaben zum Schulaustritt (wie Umzug oder obligatorische Schuldauer erreicht).

4.3.33 Elemente zum Element 'leaving'

date

optional

Datum vom Schulaustritt (wie Umzug oder obligatorische Schuldauer erreicht).

Format: `YYYY-MM-DD`

Beispiel: `2010-08-04`

schoolCity

optional

Bei einem Wegzug wird hiermit der neue Schulort definiert.

Format: `alphanumerisch`

Beispiel: `Kreuzlingen`

schoolCanton

optional

Bei einem Wegzug wird hiermit der Kanton des neuen Schulorts definiert.

Format: `alphanumerisch (2)`

Beispiel: `TG`

schoolCountry

optional

Angabe zum Land.

Format: `alphanumerisch`

Beispiel: `Schweiz`

reason

optional

Grund des Austritts.

Format: `alphanumerisch`

Beispiel: `Schweiz`

`<contacts>` *Auflistung der Eltern und Kontaktpersonen*

lehreroffice/inst/semesters/semester/schoolclasses/schoolclass/students/student/contacts

Element 'contacts' empfohlen

Dieses Element fungiert als Liste mit den entsprechenden Elementen vom Typ '<contact>'. Diese Liste umfasst die Elternangaben und Kontaktpersonen (wie Tagesmutter oder Vormund) des Schülers.

Anzahl: 1

Beispiel:

```
<contacts complete="1">  
  ...  
</contacts>
```

4.3.34 Attribute zum Element 'contacts'

`complete="1"` zwingend notwendig

Wenn der Code "1" (Standard) übergeben wird, dann ist die nachfolgende Liste mit den Kontakten vollständig. Dies bedeutet, dass alle Kontaktangaben enthalten sind, welche zum Schüler angehören. Überzählige Kontakte werden hiermit automatisch gelöscht.

Mit dem Code "0" werden die fehlenden Kontaktangaben hinzugefügt. Diese Option ist für das Nachliefern von einzelnen Kontakten sinnvoll (Push-Variante).

- 0 = Liste mit Kontaktangaben ist nicht vollständig
- 1 = Liste mit Kontaktangaben ist komplett (Standard)

Format: numerisch 0|1 (1)

Beispiel: 1

<contact>

Angaben zum Elternteil resp. Kontaktperson

lehreroffice/inst/semesters/semester/schoolclasses/schoolclass/students/student/contacts/contact

4.3.35 Element 'contact'

wenn Daten vorhanden

Datenelement für die Angaben einer Kontaktperson wie Eltern eines einzelnen Schülers.

Anzahl: 0..n

Beispiel:

```
<contact ahv="336.3449.4323.47" key="10504" >
  <prename>Beat André</prename>
  <name>Meier</name>
  ...
</contact>
```

4.3.36 Attribute zum Element 'contact'

ahv="336.3449.4323.47"

empfohlen

AHV-Nummer der Kontaktperson dient als Identifikator mit erster Priorität. Die Datenschnittstelle von CMI Lehrer-Office sucht zuerst nach einem gültigen Kontakt-Datensatz mit entsprechender AHV-Nummer. Sollte der Datensatz nicht gefunden werden, wird in zweiter Priorität nach dem Attribut 'key' gesucht (siehe nachfolgende Beschreibung).

Format: [alphanumerisch \(16\)](#)

Beispiel: 336.3449.4323.47

key="10504"

zwingend notwendig

Eindeutige ID vom Quellsystem für die Kontaktperson.

Wenn die Schulverwaltung noch keine gültigen AHV-Nummern besitzt kann in zweiter Priorität mit diesem Schlüsselwert abgeglichen werden. Sobald eine AHV-Nummer vorhanden ist, wird zuerst nach dieser gesucht.

Format: [alphanumerisch](#)

Beispiel: 10504

action=""

optional

Bearbeitungsaktion für Push-Variante, wenn die Mutationen laufend geliefert werden, denn hiermit müssen Löschaaktionen bewusst geliefert werden.

Standardmässig wird keine Aktion angegeben, da die überzähligen Kontaktangaben eines Schülers von CMI LehrerOffice ermittelt und gelöscht werden, sofern das Attribut 'complete' im Element 'contacts' mit dem Wert "1" gesetzt wurde.

Werte: [del](#) = Datensatz wird gelöscht

4.3.37 Elemente zu 'contact'

prename

zwingend notwendig

Offizieller Vorname der Kontaktperson.

Format: [alphanumerisch](#)

Beispiel: Beat André

name zwingend notwendig

Nachname der Kontaktperson.

Format: `alphanumerisch`

Beispiel: `Meier`

gender zwingend notwendig

Geschlecht der Kontaktperson.

Format: `alphanumerisch (1)`

Werte: `f` = weiblich (feminin)

`m` = männlich (masulin)

Beispiel: `m`

isGuardian zwingend notwendig

Mit dem Wert '1' wird ausgewiesen, dass diese Kontaktperson eine erziehungsberechtigte Person (gesetzliche Vertretung) des Schülers ist. Ein Schüler kann mehrere erziehungsberechtigte Personen enthalten.

Format: `0|1`

Beispiel: `1`

isParent empfohlen

Mit dem Wert '1' wird ausgewiesen, dass diese Kontaktperson die leibliche Mutter (wenn gender="f") oder der Vater (wenn gender="m") des Schülers ist. Sollte die Schuladministration diese Information nicht besitzen, darf dieses Element nicht geliefert werden, da sonst die Angaben in CMI LehrerOffice überschrieben werden.

Format: `0|1`

Beispiel: `0`

isToInform optional

Mit dem Wert '1' wird ausgewiesen, dass diese Kontaktperson Korrespondenz (wie Einladungen zu Schulbesuchen) erwünscht.

Format: `0|1`

Beispiel: `1`

isEmergencyContact optional

Mit dem Wert '1' wird ausgewiesen, dass diese Kontaktperson als Notfallkontakt des Schülers definiert ist.

Format: `0|1`

Beispiel: `1`

function empfohlen

Funktion resp. Rolle der Kontaktperson

Format: `alphanumerisch`

Beispiel: `Tagesmutter (Nachbarin)`

nativeLanguage empfohlen

Muttersprache der Kontaktperson

Format: `alphanumerisch`

Beispiel: `Spanisch`

colloquialLanguage optional

Umgangssprache der Kontaktperson

Format: [alphanumerisch](#)

Beispiel: [Englisch](#)

nationality optional

Nationalität. Ausgeschriebene Landesangabe.

Format: [alphanumerisch](#)

Beispiel: [Deutschland](#)

profession veraltet

Beruf

Format: [alphanumerisch](#)

Beispiel: [Kauffrau](#)

salutationAddress optional

Anrede für Anschrift

Format: [alphanumerisch](#)

Beispiel: [Herr](#)

salutationLetter optional

Briefanrede

Format: [alphanumerisch](#)

Beispiel: [Sehr geehrter Herr Muster](#)

aliasPrenome optional

Rufname der Kontaktperson (Vorname), welcher in allen Listen mit der Ausnahme von offiziellen Dokumenten aufgeführt wird. Personen von fremden Nationen besitzen oft mehrere, lange Vornamen, welche im alltäglichen Zusammenleben nicht verwendet werden. In der Regel ist ein Leerstring enthalten.

Format: [alphanumerisch](#)

Beispiel: [Beat](#)

aliasName optional

Deckname der Kontaktperson resp. abgekürzter Nachname, welcher in allen Listen mit der Ausnahme von offiziellen Dokumenten aufgeführt wird. Es gibt spezielle Fälle, wo gefährdete Personen so geschützt werden. In der Regel ist ein Leerstring enthalten.

Format: [alphanumerisch](#)

Beispiel: [Bond](#)

fromDate optional

Kontaktangabe ist gültig von diesem Datum (vor allem bei einem Vormund).

Format: [YYYY-MM-DD](#)

Beispiel: [2004-05-01](#)

untilDate optional

Kontaktangabe ist gültig bis zu diesem Datum.

Format: [YYYY-MM-DD](#)

Beispiel: [2007-12-31](#)

<homeAddress>

Private Wohnanschrift

lehreroffice/inst/semesters/semester/schoolclasses/schoolclass/students/student/contacts/contact/homeAddress

4.3.38 Element 'homeAddress'

empfohlen

Private Adressangaben vom Wohnsitz der entsprechenden Kontaktperson.

Anzahl: 1

Beispiel:

```
<homeAddress>
  <street>Unterdorf</street>
  <zip>8240</zip>
  <city>Thayngen</city>
  ...
</homeAddress>
```

4.3.39 Elemente zu 'homeAddress'

addressAddition

optional

Zusätzliche Adressangabe mit c/o-Angabe, Flurname oder Liegenschaftsname (für die Kantone Thurgau und Appenzell Innerrhoden).

Format: `alphanumerisch`

Beispiel: `Unteri Flur`

street **empfohlen**

Strassenangabe ohne Hausnummer. Es wird empfohlen die Strasse ausgeschreiben (nicht 'Hauptstr.' sondern mit 'Hauptstrasse' zu liefern).

Wenn in der Schulverwaltung die Angaben nicht getrennt vorliegen kann notfalls auch die Strassenangabe mit Hausnummer geliefert werden. Die Strassenangabe wird schlussendlich aus den 3 Teilen 'street', 'streetNumber' und 'streetNumberAddition' zusammengesetzt.

Format: `alphanumerisch`

Beispiel: `Unterdorf`

streetNumber

empfohlen

Hausnummer der Strassenangabe.

Format: `numerisch (4)`

Beispiel: `2`

streetNumberAddition

optional

Zusatzangabe resp. Hausteil zur Hausnummer.

Format: `alphanumerisch`

Beispiel: `b`

poBox

optional

Postfachangabe, sofern vorhanden.

Format: `alphanumerisch`

Beispiel: `Postfach 776`

zip empfohlen
Postleitzahl. Es sind maximal 5 Stellen erlaubt (Deutschland).
Format: `numerisch (5)`
Beispiel: `8240`

city empfohlen
Ortsangabe des Wohnorts.
Format: `alphanumerisch`
Beispiel: `Thayngen`

canton optional
Wohnkanton.
Format: `alphanumerisch (2)`
Beispiel: `SH`

country optional
Angabe zum Land.
Format: `alphanumerisch`
Beispiel: `Schweiz`

phone empfohlen
Private Telefonnummer vom Festnetzanschluss.
Format: `alphanumerisch`
Beispiel: `052 734 14 25`

mobile empfohlen
Private Mobil-Nummer (Natel).
Format: `alphanumerisch`
Beispiel: `078 344 15 42`

fax optional
Private Faxnummer.
Format: `alphanumerisch`
Beispiel: `052 333 30 29`

email empfohlen
Private E-Mailadresse.
Format: `alphanumerisch`
Beispiel: `hmuster@bluewin.ch`

website optional
Vollständige URL der privaten Homepage.
Format: `alphanumerisch`
Beispiel: `http://www.mywebseite.com`

`<workAddress>` *Geschäftsanschrift*

lehreroffice/inst/semesters/semester/schoolclasses/schoolclass/students/student/contacts/contact/workAddress

4.3.40 Element 'workAddress'

optional

Geschäftsadresse der entsprechenden Kontaktperson.

Anzahl: 1

Beispiel:

```
<workAddress>
  <instName>ABB Electronics</instName>
  <street>Werkstrasse</street>
  <zip>8201</zip>
  <city>Schaffhausen</city>
  ...
</workAddress>
```

4.3.41 Elemente zu 'workAddress'

instName

empfohlen

Name der Firma resp. der Institution

Format: `alphanumerisch`

Beispiel: `ABB Electronics`

instAddition

optional

Zusatzangabe der Institution

Format: `alphanumerisch`

Beispiel: `Produktionszentrum`

addressAddition

optional

Zusätzliche Adressangabe mit c/o-Angabe, Flurname oder Liegenschaftsname (für die Kantone Thurgau und Appenzell Innerrhoden).

Format: `alphanumerisch`

Beispiel: `Unteri Flur`

street

empfohlen

Strassenangabe ohne Hausnummer. Es wird empfohlen die Strasse ausgeschrieben (nicht 'Hauptstr.' sondern mit 'Hauptstrasse' zu liefern).

Wenn in der Schulverwaltung die Angaben nicht getrennt vorliegen kann notfalls auch die Strassenangabe mit Hausnummer geliefert werden. Die Strassenangabe wird schlussendlich aus den 3 Teilen 'street', 'streetNumber' und 'streetNumberAddition' zusammengesetzt.

Format: `alphanumerisch`

Beispiel: `Werkstrasse`

streetNumber

empfohlen

Hausnummer der Strassenangabe.

Format: `numerisch (4)`

Beispiel: `32`

streetNumberAddition optional

Zusatzangabe resp. Hausteil zur Hausnummer.

Format: `alphanumerisch`

Beispiel: `b`

zip optional

Postleitzahl der Behörde resp. der Schulgemeinde. Es sind maximal 5 Stellen erlaubt (Deutschland).

Format: `numerisch (5)`

Beispiel: `8201`

city empfohlen

Ortsangabe des Geschäfts resp. der Schulgemeinde.

Format: `alphanumerisch`

Beispiel: `Schaffhausen`

canton optional

Kantonsangabe. Bei Adressen aus dem Ausland wird dieses Element leer erwartet.

Format: `alphanumerisch (2)`

Beispiel: `SH`

country optional

Angabe zum Land.

Format: `alphanumerisch`

Beispiel: `Schweiz`

phone empfohlen

Persönliche Telefonnummer (Direktwahl).

Format: `alphanumerisch`

Beispiel: `052 340 30 39`

phoneMain optional

Haupttelefonnummer (Zentrale oder vom Sekretariat).

Format: `alphanumerisch`

Beispiel: `052 340 30 20`

mobile optional

Mobil-Nummer vom Geschäft (Natel).

Format: `alphanumerisch`

Beispiel: `076 143 22 91`

fax optional

Faxnummer.

Format: `alphanumerisch`

Beispiel: `052 340 30 21`

email optional

Geschäftliche E-Mailadresse.

Format: `alphanumerisch`

Beispiel: `hans.muster@abb.ch`

website

optional

Vollständige URL der Homepage der Firma/Behörde

Format: `alphanumerisch`

Beispiel: `http://www.webseite.com`

<code><history></code>	<i>Schülergeschichte mit alten Wohnadressen</i>
------------------------------	---

lehreroffice/inst/semesters/semester/schoolclasses/schoolclass/students/student/history

4.3.42 Element 'history'

optional

Dieses Element fungiert als Liste mit den entsprechenden Elementen vom Typ '<addressitem>'. In einer späteren Version sind andere Typen von Einträgen möglich. Diese Liste umfasst die Schülergeschichte wie vorangehende Wohnadressen.

Anzahl: 1

Beispiel:

```
<history complete="1">
  ...
</history>
```

4.3.43 Attribute zum Element 'history'

complete="1"

zwingend notwendig

Wenn der Code "1" (Standard) übergeben wird, dann ist die nachfolgende Liste vollständig.

- | | | |
|---|---|-------------------------------|
| 0 | = | Liste ist nicht vollständig |
| 1 | = | Liste ist komplett (Standard) |

Format: numerisch 0|1 (1)

Beispiel: 1

<code><addressItem></code>	<i>Frühere Wohnadresse</i>
----------------------------------	----------------------------

lehreroffice/inst/semesters/semester/schoolclasses/schoolclass/students/student/history/addressItem

4.3.44 Element 'addressitem'

optional

Alte Wohnadresse des Schülers.

Anzahl: 0..n

Beispiel:

```
<addressItem>
  <street>Unterdorf</street>
  <zip>8200</zip>
  <city>Schaffhausen</city>
  <canton>SH</canton>
  ...
</addressItem>
```

4.3.45 Elemente zu 'addressitem'

street empfohlen

Strassenangabe ohne Hausnummer. Es wird empfohlen die Strasse ausgeschreiben (nicht 'Hauptstr.' sondern mit 'Hauptstrasse' zu liefern).

Wenn in der Schulverwaltung die Angaben nicht getrennt vorliegen kann notfalls auch die Strassenangabe mit Hausnummer geliefert werden. Die Strassenangabe wird schlussendlich aus den 3 Teilen 'street', 'streetNumber' und 'streetNumberAddition' zusammengesetzt.

Format: `alphanumerisch`

Beispiel: `Unterdorf`

streetNumber empfohlen

Hausnummer der Strassenangabe.

Format: `numerisch (4)`

Beispiel: `2`

streetNumberAddition optional

Zusatzangabe resp. Hausteil zur Hausnummer.

Format: `alphanumerisch`

Beispiel: `b`

zip empfohlen

Postleitzahl. Es sind maximal 5 Stellen erlaubt (Deutschland).

Format: `numerisch (5)`

Beispiel: `8200`

city empfohlen

Ortsangabe

Format: `alphanumerisch`

Beispiel: `Schaffhausen`

canton empfohlen

Kantonsangabe

Format: [alphanumerisch](#)

Beispiel: [SH](#)

country optional

Land

Format: [alphanumerisch](#)

Beispiel: [Schweiz](#)

fromDate optional

Wohnadresse war gültig von diesem Datum.

Format: [YYYY-MM-DD](#)

Beispiel: [2004-05-01](#)

untilDate empfohlen

Wohnadresse war gültig bis zu diesem Datum.

Format: [YYYY-MM-DD](#)

Beispiel: [2007-12-31](#)

<code><schoolData></code>	<i>Schulform (bei integrierten Schülern)</i>
---------------------------------	--

lehreroffice/inst/semesters/semester/schoolclasses/schoolclass/students/student/schoolData

4.3.46 Element 'schoolData'

empfohlen

Schülerspezifische Angaben zur Schulform.

Anzahl: 1

Beispiel:

```
<schoolData>
  <schoolYears>8</schoolYears>
  <schoolForm>0</schoolForm>
  ...
</schoolData>
```

4.3.47 Elemente zu 'schoolData'

schoolYears

empfohlen

Anzahl besuchte Schuljahre, wobei das laufende Schuljahr mitgezählt wird. Somit erhält ein Schüler der 2. Primarklasse den Wert '2', ein Schüler der 2. Sekundarklasse den Wert '8'. Bei Kantonen, wo der Kindergarten obligatorisch ist, werden die Kindergartenjahre für die Schuljahre mitgezählt.

Format: numerisch (2)

Beispiel: 8

schoolQuota

optional

Schüler besucht das Pensum gemäss dieser Jahresangabe (bei integrierten Schülern). Das obige Element 'schoolYears' hingegen weist die absoluten Schuljahre des Schülers aus.

Wenn z.B. ein Schüler im 8. Schuljahr (schoolYears) das Pensum der 1. Realklasse besucht, enthält dieses Element somit den Wert "1" und 'schoolYears' den Wert "7". Dieses Element wird in der Regel im Zusammenhang mit 'schoolForm' verwendet.

Dieser Wert entspricht dem Programmjahr vom Feld 'ctSchYear' vom Bundesamt für Statistik. Mit Programmjahr wird der Unterricht eines Schuljahres bezeichnet. Auf der Sekundarstufe II ist der Begriff „Lehrjahr“ geläufiger, auf der Stufe obligatorische Schule spricht man auch von 'Klasse'.

Format: numerisch (1)

Beispiel: 1

schoolForm optional

Schulart bei integrierten Schülern, welche eine Regelklasse besuchen. Die Werte entsprechen der Liste wie beim Element 'classForm' vom Abschnitt 'classDefinition'.

Werte:	KG	=	Kindergarten
	BS	=	Basisstufe
	ES	=	Einschulungsklasse
	KK[n]	=	Kleinklasse A..D
	O	=	Sonderklasse
	P	=	Primarklasse
	S[n]	=	Sekundarklasse (Abteilung, Niveau als zweites Zeichen z.B. "SA")
	R	=	Realklasse
	W	=	Werkklasse

Beispiel: O

schoolFormCode optional

BFS-Code der Schulart

Format: alphanumerisch

Beispiel: 11

schoolFormName optional

Vollständiger Name der Schulart

Format: alphanumerisch

Beispiel: Kindergarten

DAZ veraltet

Deutsch als Zweitsprache. Dieses Element ist nur noch zur Abwärtskompatibilität vorhanden. Bitte künftig die Angabe unter 'individualFostering' liefern (siehe nachfolgender Abschnitt).

Format: 0|1

VM veraltet

Verstärkte Massnahmen im Einzelfall des Schülers. Dieses Element ist nur noch zur Abwärtskompatibilität vorhanden. Bitte künftig die Angabe unter 'individualFostering' liefern (siehe nachfolgender Abschnitt).

Format: 0|1

planStatus optional

Schüler/innen mit besonderen Bedürfnissen werden in unterschiedlichen Schulprogrammen unterrichtet: in einer Sonderschule, in einer Sonderklasse oder integriert in einer Regelklasse (integrative Schulungsform, integrierte Heilpädagogik). Integrierte Schüler/innen unterscheiden sich von ihren Klassenkamerad/innen dadurch, dass sie nicht nach dem Regellehrplan, sondern nach einem individuellen Lehrplan unterrichtet werden.

Das Merkmal Lehrplanstatus ist nur für Schulprogramme mit besonderem Lehrplan zu erfassen. Auf den anderen Schulstufen wird der Standardwert 10 eingesetzt. Die Nomenklatur wurde vom Bundesamt für Statistik übernommen (siehe Lehrplanstatus D.5.4).

Werte:	10	=	Regellehrplan
	20	=	Teilweise individuelle Lernziele (1 bis 2 Lektionen)
	30	=	Mehrheitlich individuelle Lernziele (in 3 Lektionen und mehr)

Beispiel: 10

preparationHighschool

optional

Wenn der Schüler den Vorbereitungsunterricht für eine Höhere Schule wie Berufsmaturität besucht, wird dies mit diesem Element gekennzeichnet.

Werte: **BMT** = Berufsmaturität technische Richtung
BMK = Berufsmaturität kaufmännische Richtung
BMG = Berufsmaturität gestalterische Richtung
BMW = Berufsmaturität gewerbliche Richtung
BMN = Berufsmaturität naturwissenschaftliche Richtung
BMO = Berufsmaturität gesundheitliche und soziale Richtung

Beispiel: **BMT**

teachingForm

optional

Für die Vorschule, die Primarstufe und die Sekundarstufe I erübrigt sich in der Regel die Unterscheidung der Ausbildungsform, da es sich um schulische Vollzeitausbildungen handelt (es wird der Standardwert 10 eingesetzt).

Für die Sekundarstufe II und die Tertiärstufe werden die Merkmalsausprägungen wie nachfolgend definiert. Die Nomenklatur wurde vom Bundesamt für Statistik übernommen (siehe Ausbildungsform D.5.3).

Werte: **10** = Schulische Vollzeitausbildung
20 = Duale berufliche Grundbildung
30 = Schulische Teilzeitausbildung

Beispiel: **10**

teachingProfession

optional

Bei Berufsschulen wird der Lehrberuf als Berufsbezeichnung des Schülers geliefert. Bei Oberstufen kann in diesem Feld der gewählte Beruf erfasst werden.

Format: **alphanumerisch**

Beispiel: **Schreiner**

teachingCompanyKey

optional

Bei Berufsschulen wird die zuständige Lehrfirma des Schülers geliefert. Verknüpfung über den Schlüsselwert zu einer bestehenden Adresse einer Lehrfirma.

Format: **alphanumerisch**

Beispiel: **67044**

teachingCompanyKey

optional

Bei Berufsschulen wird die zuständige Lehrfirma des Schülers geliefert. Verknüpfung über den Schlüsselwert zu einer bestehenden Adresse einer Lehrfirma.

Format: **alphanumerisch**

Beispiel: **67044**

<individualFostering>

Fördermassnahmen

lehreroffice/inst/semesters/semester/schoolclasses/schoolclass/students/student/schoolData/individualFostering

4.3.48 Element 'individualFostering'

optional

Schülerspezifische Fördermassnahmen. Die Auswahl der Fördermassnahmen ist je nach Kanton unterschiedlich und auf folgender Seite weiter dokumentiert:

<http://www.lehreroffice.ch/wiki/doku.php?id=lehreroffice:desktop:module:schullaufbahn:foerderangebote>

Die einzelnen Förderangebote sind im XML-Schema bewusst mit den gängigen Abkürzungen (in deutscher Sprache) geführt. Jedes Element besitzt das Attribut 'active', womit die Fördermassnahme pro Schüler mit dem Wert '1' ausgewählt werden kann. Langfristig ist es möglich, dass einzelne Fördermassnahmen mit weiteren Unterelementen ausgebaut werden.

Anzahl: 1

Beispiel:

```
<individualFostering>
  <bega active=1>
    <description>Fussball</description>
  </bega>
  <daz active=1/>
  <logo active=1>
    <occurrences>
      <occurrence key=1201>
        <occurrenceType>Massnahme</occurrenceType>
        <date>2019-08-12</date>
        <untildate>2020-01-31</untildate>
        <lessonsperweek>2</lessonsperweek>
      </occurrence>
    </occurrences>
  </logo>
  <dysk active=0>
    <occurrences>
      <occurrence key=1248>
        <occurrenceType>Abklärung</occurrenceType>
        <date>2019-09-13</date>
      </occurrence>
    </occurrences>
  </dysk>
  ...
</individualFostering>
```

4.3.49 Elemente zu 'individualFostering'

bega

optional

Begabungsförderung (Bega). Mit dem Unterelement 'description' kann eine detailliertere Angabe zur Begabtenförderung wie 'Fussball' in Textform mitgeliefert werden (siehe obiges Beispiel).

besl	optional
Beschleunigung. Dieses Förderangebot steht nur für den Kanton Solothurn zur Verfügung.	
daz	optional
Deutsch als Zweitsprache (DaZ).	
dysk	optional
Dyskalkulie (Dysk).	
ffsp	optional
Frühfremdsprache für Zugezogene. Dieses Förderangebot steht nur für den Kanton Solothurn zur Verfügung.	
hpf	optional
Heilpädagogische Früherziehung (HPF). Dieses Förderangebot steht zurzeit nur für die Kantone Graubünden und Solothurn zur Verfügung.	
if (Kanton GR: ifml, ifol)	optional
Integrative Förderung oder integrierte Heilpädagogik. Je nach Kanton sind unterschiedliche Bezeichnungen im Umlauf:	
Aargau:	Integrierte Heilpädagogik
Graubünden:	Integrative Förderung (IF oL, mL)
Schaffhausen:	Integrative Schulform
Solothurn:	Schulische Heilpädagogik
Zürich:	Integrative Förderung
ifp	optional
Integrative Förderung als Prävention (IF P). Dieses Förderangebot steht nur für den Kanton Graubünden zur Verfügung.	
is	optional
Integrierte Sonderschulung (IS). Je nach Kanton sind auch hier unterschiedliche Bezeichnungen im Umlauf:	
Solothurn: ISM §37	
lega	optional
Legasthenie (Lega).	
logo	optional
Logopädie (Logo).	
psymo	optional
Psychomotorik (Psymo).	
psyth	optional
Psychotherapie. Diese Fördermassnahme wird von Seiten CMI LehrerOffice noch nicht unterstützt.	
verl	optional
Verlangsamung. Dieses Förderangebot steht nur für den Kanton Solothurn zur Verfügung.	

vm

optional

Verstärkte Massnahmen im Einzelfall des Schülers. Diese Fördermassnahme ersetzt die bisherige Bezeichnung „unterstützende Massnahmen im Einzelfall“ (UME). Dieses Förderangebot steht nur für den Kanton Aargau zur Verfügung.

4.3.50 Element 'occurrences'

wenn Daten vorhanden

Element 'occurrences'

Dieses Element fungiert als Liste mit den entsprechenden Elementen vom Typ '<occurrence>'. Die Liste umfasst Abklärungen und Massnahmen zu einem der Förderangebote.

Anzahl: 1

Beispiel:

```
<occurrences complete="1">  
  ...  
</occurrences>
```

4.3.51 Attribute zum Element 'occurrences'

complete="1"

zwingend notwendig

Wenn der Code "1" (Standard) übergeben wird, dann ist die nachfolgende Liste vollständig.

0 = Liste ist nicht vollständig

1 = Liste ist komplett (Standard)

Format: numerisch 0|1 (1)

Beispiel: 1

4.3.52 Attribute zum Element 'occurrence'

key="10504"

zwingend notwendig

Eindeutige ID vom Quellsystem für die Abklärung bzw. Massnahme.

Format: alphanumerisch

Beispiel: 10504

action=""

optional

Bearbeitungsaktion für Push-Variante, wenn die Mutationen laufend geliefert werden, denn hiermit müssen Löschaaktionen bewusst geliefert werden.

Standardmässig wird keine Aktion angegeben.

Werte: del = Datensatz wird gelöscht

4.3.53 Elemente zum Element 'occurrence'

date

optional

Datum des Zeitpunkts, wann die Abklärung stattgefunden hat bzw. eine Massnahme startet.

Format: YYYY-MM-DD

Beispiel: 2019-11-04

occurrenceType zwingend notwendig

Die Art des Ereignisses wie Abklärung oder Massnahme. Je nach Kanton kann als Attribut 'subtype' ein Text mitgegeben werden, über den die Massnahme noch näher definiert ist.

Werte: [Abklärung](#)
[Massnahme](#)

Attribut-Werte: [ohne ILZ](#) = Massnahme ohne ILZ (bei IF im Kanton Luzern)
[mit ILZ](#) = Massnahme mit ILZ (bei IF im Kanton Luzern)

[Anfangsunterricht](#) (bei DaZ im Kanton Luzern)
[Aufbauunterricht](#) (bei DaZ im Kanton Luzern)
[Asyl](#) (bei DaZ im Kanton Luzern)

Beispiel: [Massnahme](#)

remarks optional

Allgemeine Informationen zur Abklärung bzw. Massnahme.

Format: [alphanumerisch](#)

Beispiel: [Die Abklärung fand auf Wunsch der Lehrperson mit Einwilligung der Eltern statt.](#)

untildate optional

Datum bis wann eine Massnahme durchgeführt werden soll.

Format: [YYYY-MM-DD](#)

Beispiel: [2019-11-04](#)

lessonsperweek optional

Anzahl Wochenlektionen bei einer Massnahme.

Format: [numerisch](#)

Beispiel: [2](#)

lessonstotal optional

Gesamtzahl bewilligter Lektionen bei einer Massnahme.

Format: [numerisch](#)

Beispiel: [60](#)

<subjects>

Auflistung der Schulfächer

lehreroffice/inst/semesters/semester/schoolclasses/schoolclass/students/student/sub-
jects

4.3.54 Element 'subjects'

optional

Dieses Element fungiert als Liste mit den entsprechenden Elementen vom Typ '<subject>'. Diese Liste umfasst alle Niveau- und Wahlfächer, welche dieser Schüler besucht.

Anzahl: 1

Beispiel:

```
<subjects complete="1">  
  ...  
</subjects>
```

4.3.55 Attribute zum Element 'subjects'

complete="1"

zwingend notwendig

Wenn der Code "1" (Standard) übergeben wird, dann ist die nachfolgende Schulfächerliste vollständig. Dies bedeutet, dass alle Niveau- und Wahlfächer enthalten sind, welche der Schüler besucht. Überzählige Fächer werden hiermit automatisch entfernt, welche bei Bedarf jederzeit mit den aktuellen Beurteilungen ohne Datenverluste wieder reaktiviert werden können.

Mit dem Code "0" werden die fehlenden Schulfachzuteilungen hinzugefügt. Diese Option ist für das Nachliefern von einzelnen Fachzuteilungen sinnvoll (Push-Variante).

0	=	Fächerliste ist nicht vollständig
1	=	Fächerliste ist komplett (Standard)

Format: numerisch 0|1 (1)

Beispiel: 1

ilzComplete="1" optional

Bei einem einzelnen Schulfach kann man mit dem Element 'ILZ' das Flag 'individuelle Lernziele' setzen oder (mit dem Wert '0') auch wieder entfernen. Sofern man die Information, bei welchen Schulfächern eine Schülerin/ein Schüler keine individuellen Lernziele mehr hat, nicht liefern kann, kann die Entfernung von ILZ-Fächern über dieses Attribut gesteuert werden. Mit 'ilzComplete="1"' werden somit alle nicht gelieferten Zuteilungen entfernt.

0	=	Es werden keine ILZ-Zuteilungen entfernt, sofern beim Fach das Element 'ILZ' nicht explizit mit '0' geliefert wird (Standard).
1	=	Um ILZ-Zuteilungen zu entfernen, ohne das entsprechende Fach mit dem Element 'ILZ'='0' liefern zu müssen, kann ilzComplete="1" gesetzt werden.

Format: numerisch 0|1 (1)

Beispiel: 1

`<subject>` *Fach- und Niveauzuteilung des Schülers*

lehreroffice/inst/semesters/semester/schoolclasses/schoolclass/students/student/subjects/subject

4.3.56 Element 'subject'

wenn Daten vorhanden

Datenelement eines einzelnen Schulfaches, das durch den aktuellen Schüler besucht wird.

Anzahl: 0..n

Beispiel:

```
<subject key="D" caption="Deutsch">
  ...
</subject>
```

4.3.57 Attribute zum Element 'subject'

key="D"

zwingend notwendig

Eindeutige und statische ID vom Quellsystem für das Schulfach, welche über alle Schuljahre unverändert bleibt.

Format: *alphanumerisch*

Beispiel: D

In CMI LehrerOffice werden die Schulfächer einmalig definiert und einmalig mit dem dazugehörigen Schlüssel der Schulverwaltung ergänzt.

Bei den Fachdefinitionen darf der Wert im Feld 'Schlüssel' von CMI LehrerOffice nicht verändert werden, da sonst der Zeugnisdruck nicht mehr korrekt funktionieren wird! Für den Abgleich wird ein zweiter, separater Schlüsselwert von CMI LehrerOffice zur Verfügung gestellt, der frei an die Schulverwaltung angepasst werden darf.

caption="Deutsch"

empfohlen

Bezeichnung des Schulfachs, damit der gelieferte Fachschlüssel in CMI LehrerOffice am richtigen Ort eingetragen werden kann.

Format: *alphanumerisch*

Beispiel: *Deutsch*

4.3.58 Elemente zu 'subject'

niveau

empfohlen

Niveauangabe bei einem Niveaufach. Fächer ohne Niveauzuteilung müssen dieses Element nicht besitzen. Wenn bei einem gesetzten Niveaufach die Fachangabe entfernt werden soll, dann muss ein Leerstring geliefert werden.

Format: *alphanumerisch*

Beispiel: B

group

optional

Zusatzangabe zum Niveau (für dynamische Gruppeneinteilung).

Format: *alphanumerisch*

Beispiel: 1

addition

optional

Zusatzangabe zum Fach (wie Instrument bei Musikunterricht oder Sprache bei HKS).

Format: `alphanumerisch`

Beispiel: `Trompete`

ILZ

empfohlen

Mit dem Wert '1' wird ausgewiesen, dass dieser Schüler in diesem Fach individuelle Lernziele (ILZ) hat.

Format: `0|1`

Beispiel: `1`

`<absences>` *Absenz*

lehreroffice/inst/semesters/semester/schoolclasses/schoolclass/students/student/absences/

4.3.59 Element 'absences'

wenn Daten vorhanden

Dieses Element fungiert als Liste mit den Elementen vom Typ '`<absence>`'. Die Liste umfasst die Abwesenheiten des Schülers.

Anzahl: 1

Beispiel:

```
<absences complete="1">
  ...
</absences>
```

4.3.60 Attribute zum Element 'absences'

`complete="1"`

zwingend notwendig

Wenn der Code "1" (Standard) übergeben wird, dann ist die nachfolgende Liste vollständig.

0 = Liste ist nicht vollständig

1 = Liste ist komplett (Standard)

Format: `numerisch 0|1 (1)`

Beispiel: 1

4.3.61 Element 'absence'

empfohlen

Pro Abwesenheit gibt es ein Element 'absence'. Das Element wird anhand eines eindeutigen Schlüssels identifiziert.

Anzahl: `0..n`

Beispiel:

```
<absence key="10504">
  <absenceType>Krank/Unfall</absenceType>
  <date>2019-11-04</date>
  ...
  <duration unit="Lektion">5</duration>
  ...
</absence>
```

4.3.62 Attribute zum Element 'absence'

`key="10504"`

zwingend notwendig

Eindeutige ID vom Quellsystem für die Absenz.

Format: `alphanumerisch`

Beispiel: 10504

action=""

optional

Bearbeitungsaktion für Push-Variante, wenn die Mutationen laufend geliefert werden, denn hiermit müssen Löschaaktionen bewusst geliefert werden.

Standardmässig wird keine Aktion angegeben.

Werte: `del` = Datensatz wird gelöscht

4.3.63 Elemente zu 'absence'

absenceType

zwingend notwendig

Mit diesem Element wird die Unterkategorie einer Absenz angegeben. Die möglichen Werte sind vorgegeben. Mit den Typen 'Schulspezifisch1' und 'Schulspezifisch2' kann eine Schule noch zwei eigene Kategorien von Absenzen definieren und in der Datenbank festlegen, ob diese Kategorie zu den entschuldigten Absenzen gezählt werden soll oder nicht.

Werte: `Unentschuldigt`
`Krank/Unfall`
`Jokertage`
`Dispens`
`Schnuppertage`
`Schulspezifisch1`
`Schulspezifisch2`

Beispiel: `Krank/Unfall`

date

empfohlen

Datum des Zeitpunkts der Absenz. Ist eine Abwesenheit länger als ein Tag, wird das Startdatum angegeben.

Format: `YYYY-MM-DD`

Beispiel: `2019-11-04`

remarks

optional

Eine Anmerkung zur Absenz. In diesem Textfeld können weitere Details zur Absenz festgehalten werden.

Format: `alphanumerisch`

Beispiel: `Der Schüler ist von der Mutter telefonisch für den Vormittag entschuldigt worden.`

duration

empfohlen

Die Dauer einer Absenz. Mit dem Attribut 'unit' wird die Einheit (Lektion, Halbtage oder Tag) angegeben. Das Attribut muss zwingend mitgeliefert werden.

Format: `numerisch`

Beispiel: `5`

isOpen

zwingend notwendig

Mit dem Wert '0' wird der Journaleintrag als 'Erledigt' ausgewiesen, mit dem Wert '1' als noch offener Eintrag.

Format: `0|1`

Beispiel: `0`

`<excused>` *Entschuldigung für eine Absenz*

lehreroffice/inst/semesters/semester/schoolclasses/schoolclass/students/student/absences/absence/excused

4.3.64 Element 'excused'

optional

Über das Element 'excused' kann in CMI LehrerOffice zu einer Absenz das Datum des Eingangs der Entschuldigung sowie eine Bemerkung erfasst werden.

Anzahl: 1

Beispiel:

```
<excused>  
  <excusedDate>2019-12-01</excusedDate>  
  <excusedReason>krank (Übelkeit und Erbrechen)</excusedReason>  
</excused>
```

4.3.65 Elemente zu 'excused'

excusedDate

zwingend notwendig

Datum des Zeitpunkts des Eingangs der Entschuldigung.

Format: YYYY-MM-DD

Beispiel: 2019-11-04

excusedReason

optional

Über dieses Textfeld kann der Grund für die Absenz bei der offiziellen Entschuldigung noch detaillierter ausgeführt werden.

Format: alphanumerisch

Beispiel: krank (Übelkeit und Erbrechen)

5 Definitionen aller Gruppen (Fachklassen / Niveaunklassen)

Mit dieser Methode können die Definitionen aller manuellen Gruppen (Fachklassen, Niveaunklassen) übernommen werden. Diese Angaben werden schlussendlich im Modul 'Gruppen' von CMI LehrerOffice abgelegt werden.

Definitionen aller Gruppen

- Gruppendefinitionen
- Grunddaten der Gruppen

SOAP-Methode: `LO_getStudyGroupDefinitions`

Parameter: `int semesterYear, int semesterPart`

Dateiname: `"SJ" + Schuljahr + "-" + Semester + "_Gruppendefinitionen.xml"`

Beispiel: `SJ2011-2_Gruppendefinitionen.xml`

5.1 SOAP – Anfrage für Definitionen aller Gruppen

<LO_getStudyGroupDefinitions>

Methodenaufruf mit Parameter

soap:Envelope/soap:Body/LO_getStudyGroupDefinitions

5.1.1 Element 'LO_getStudyGroupDefinitions'

zwingend notwendig

Dieses Element ist der eigentliche Methodenaufruf für die Abfrage/Aktion der Definitionen der Gruppen. Jede Schulverwaltungslösung kann diese SOAP-Methode als Web-Service zur Verfügung stellen. Das erwartete Resultat als XML ist im nachfolgenden Abschnitt dokumentiert.

Beispiel:

```
<soap:Body>
  <LO_getStudyGroupDefinitions xmlns="http://www.lehrer-
office.ch/xmlns/soap/2010/exchange-v1">
    <semesterYear>2010</semesterYear>
    <semesterPart>1</semesterPart>
  </LO_getStudyGroupDefinitions>
</soap:Body>
```

5.1.2 Attribute zum Element 'LO_getStudyGroupDefinitions'

xmlns="http://www.lehreroffice.ch/xmlns/soap/2010/exchange-v1"

zwingend notwendig

Dieser Namespace muss als Standardnamespace, also ohne Namespace-Präfix angegeben werden.

Diese Versionsangabe hat keinen Zusammenhang mit der Programmversion von CMI LehrerOffice als Anwendung.

Format: [URL](#)

Beispiel: <http://www.lehreroffice.ch/xmlns/soap/2010/exchange-v1>

5.1.3 Elemente zu 'LO_getStudyGroupDefinitions' als Parameter

`semesterYear`

zwingend notwendig

Vierstellige Ganzzahl für das Schuljahr, wobei das Kalenderjahr des Schulanfangs geliefert wird. Für das Schuljahr 2010/11 wird somit die Zahl '2010' geliefert.

Format: `numerisch (4)`

Beispiel: `2010`

`semesterPart`

zwingend notwendig

Einstellige Ganzzahl für die Semesterangabe.

Format: `1|2`

Beispiel: `1`

5.2 Definitionen aller Gruppen – XML-Struktur

<code><lehreroffice></code>	Dateninhalt
<code><inst></code>	Institution / Schulgemeinde
<code><source></code>	Angaben zum Quellsystem
<code></source></code>	
<code><semesters></code>	Auflistung der Semester
<code><semester></code>	Semesterdaten
<code><studygroups></code>	Auflistung der Arbeitsgruppen
<code><studygroup></code>	Arbeitsgruppe
<code><groupDefinition></code>	Gruppendefinition
<code></groupDefinition></code>	
<code><groupData></code>	Stammdaten der Arbeitsgruppe
<code></groupData></code>	
<code></studygroup></code>	
<code></studygroups></code>	
<code></semester></code>	
<code></semesters></code>	
<code></inst></code>	
<code></lehreroffice></code>	

5.3 Definitionen aller Gruppen – Elemente

<code><lehreroffice></code>	Dateninhalt
<code><inst></code>	Institution / Schulgemeinde
<code><source></code>	Angaben zum Quellsystem
<code></source></code>	
<code><semesters></code>	Auflistung der Semester
<code><semester></code>	Semesterdaten
<code></semester></code>	
<code></semesters></code>	
<code></inst></code>	
<code></lehreroffice></code>	

Obige Elemente und dessen Attribute und Unterelemente sind im vorangehenden Kapitel ' Daten für Lehrpersonen – Elemente' detailliert dokumentiert. Nachfolgend sind nun die Elemente innerhalb vom Element '<semester>' beschrieben.

`<studyGroups>` *Auflistung der Gruppen*

lehreroffice/inst/semesters/semester/studyGroups

5.3.1 Element 'studyGroups'

zwingend notwendig

Dieses Element fungiert als Liste mit Unterelementen vom Typ '<studyGroup>'. Hier müssen alle Definitionen für alle Gruppen geliefert werden.

Anzahl: 1

Beispiel:

```
<studyGroups complete="1">  
  <studyGroup key="sp_1">  
    ...  
  </studyGroup>  
  <studyGroup key="sp_2">  
    ...  
  </studyGroup>  
  ...  
</studyGroups>
```

5.3.2 Attribute zum Element 'studyGroups'

`complete="1"`

zwingend notwendig

Wenn der Code "1" übergeben wird, dann ist die nachfolgende Liste mit den Gruppen vollständig. Dies bedeutet, dass alle Gruppen enthalten sind. Überzählige Gruppen werden hiermit automatisch innerhalb CMI LehrerOffice ausgeblendet.

Mit dem Code "0" (Standard) werden die fehlenden Gruppen hinzugefügt. Diese Option ist für das Nachliefern von einzelnen Gruppen sinnvoll (Push-Variante).

- 0 = Liste mit Gruppen ist nicht vollständig
- 1 = Liste mit Gruppen ist komplett

Format: `numerisch 0|1 (1)`

Beispiel: 1

<code><studyGroup></code>	<i>Arbeitsgruppe</i>
---------------------------------	----------------------

lehreroffice/inst/semesters/semester/studyGroups/studyGroup

5.3.3 Element 'studyGroup'

zwingend notwendig

Definition einer Arbeitsgruppe (Niveaunklasse, Fachklasse). Es müssen alle Gruppen in einer XML-Datei geliefert werden.

Anzahl: 1..n

Beispiel:

```
<studyGroup key="SHA-E2" id="19" modDateTime="2010-01-03T09:32:59">
  ...
</studyGroup>
```

5.3.4 Attribute zum Element 'studyGroup'

key="SHA-E2"

zwingend notwendig

Eindeutige und statische ID vom Quellsystem für die Arbeitsgruppe, welche über alle Schuljahre unverändert bleibt.

Format: alphanumerisch

Beispiel: SHA-E2

In CMI LehrerOffice werden die Gruppengefässe einmalig definiert, wobei dazu für jede Arbeitsgruppe das dazugehörige Schulfach und Niveau hinterlegt wird. Die einzelnen Schüler und die entsprechenden Lehrpersonen werden am Ende eines Schuljahres nun von der aktuellen Gruppeneffinition zur nächsten weitergeschoben. Vergleichbar mit einem Klassenzimmer, das immer wieder mit neuen Schülern und Lehrpersonen belegt wird.

id

optional

Bei Schulverwaltungslösungen (wie EcoOpen, Sclaris), welche für jede Schulklasse resp. Gruppe eine neue ID zu Beginn der Schulzeit generieren, kann die laufende ID in CMI LehrerOffice auch als zweites Suchkriterium genutzt werden.

Schulverwaltungslösungen (wie iCampus), welche dasselbe System mit den fixen Schlüsselwerten wie CMI LehrerOffice anwendet, darf dieses Attribut nicht nutzen.

Format: alphanumerisch

Beispiel: 19

modDateTime

optional

Modifikationsdatum im XSD-Format. Letzte Mutation des Datensatzes auf dem Quellsystem. Wenn dieser Zeitstempel älter oder gleich dem abgespeicherten TimeStamp in CMI LehrerOffice ist, wird dieses Element und alle Unterelemente nicht verarbeitet. Dank diesem TimeStamp kann die Verarbeitungszeit stark optimiert werden.

Auch bei einer Änderung der dazugehörigen Lehrpersonen, der untergeordneten Schülerzuteilungen muss ein geänderter Zeitstempel geliefert werden!

Format: YYYY-MM-DDTHH:MM:SS

Beispiel: 2010-01-03T09:32:59

active

optional

Arbeitsgruppe ist in diesem Semester aktiv (1) gekennzeichnet oder nicht (0).

Format: `numerisch 0|1 (1)`

Beispiel: `1`

<groupDefinition>

Gruppendefinition

lehreroffice/inst/semesters/semester/studyGroups/studyGroup/groupDefinition

5.3.5 Element 'groupDefinition'

(nicht implementiert) optional

Damit künftig Gruppengefässe automatisch in CMI LehrerOffice erstellt werden können, kann die Definition der Arbeitsgruppe mit diesen Angaben einmalig erstellt werden, wenn die Gruppe gemäss Schlüssel nicht gefunden werden konnte. Diese Definitionen sind im Gegensatz zu 'groupData' nicht semesterabhängig.

Anzahl: 1

Beispiel:

```
<groupDefinition>
  ...
</groupDefinition>
```

5.3.6 Attribute zu 'groupDefinition'

modDateTime

optional

Modifikationsdatum im XSD-Format. Letzte Mutation des Datensatzes auf dem Quellsystem. Wenn dieser Zeitstempel älter oder gleich dem abgespeicherten TimeStamp in CMI LehrerOffice ist, wird dieses Element und alle Unterelemente nicht verarbeitet. Dank diesem TimeStamp kann die Verarbeitungszeit stark optimiert werden.

Dieser Zeitstempel gilt nur für die Gruppendefinition und nicht für die folgenden Schülerzuteilungen.

Format: YYYY-MM-DDTHH:MM:SS

Beispiel: 2010-01-03T09:32:59

5.3.7 Elemente zu 'groupDefinition'

name

zwingend notwendig

Allgemeingültige Gruppenbezeichnung als Kürzel, wobei semesterabhängige Angaben und Namen von Lehrpersonen zu unterlassen sind.

Format: alphanumerisch

Beispiel: Englisch II

unitKey

optional

Grosse Schulgemeinden können in Schuleinheiten unterteilt werden, wobei jede Arbeitsgruppe einer Schuleinheit zugeordnet wird.

Eindeutige ID vom Quellsystem für die Schuleinheit. Die Schuleinheiten müssen vorab innerhalb von CMI LehrerOffice definiert und mit demselben Schlüsselwert wie im Quellsystem eingestellt werden.

Format: alphanumerisch

Beispiel: OZ1

unitName optional

Name der Schuleinheit. Dient lediglich als Information für die Zuordnung der Schuleinheit zur ID, wird nicht übernommen.

Format: [alphanumerisch](#)

Beispiel: [Sekundarschule](#)

bur="72765790" empfohlen

Zur Identifikation wird die BUR-Nummer SSP vom Amt für Statistik verwendet. Diese BUR-Nummer SSP ist ein eindeutiger Identifikator der Bildungsinstitute der Schweiz.

Format: [numerisch \(20\)](#)

Beispiel: [72765790](#)

subjectKey optional

Eindeutige und statische ID vom Quellsystem für das Schulfach, welche über alle Schuljahre unverändert bleibt. Wenn das Schulfach in CMI LehrerOffice und diese Angabe nicht übereinstimmt, wird eine Warnung ausgegeben.

Format: [alphanumerisch](#)

Beispiel: [D](#)

In CMI LehrerOffice werden die Schulfächer einmalig definiert und einmalig mit dem dazugehörigen Schlüssel der Schulverwaltung ergänzt.

Bei den Fachdefinitionen darf der Wert im Feld 'Schlüssel' von CMI LehrerOffice nicht verändert werden, da sonst der Zeugnisdruck nicht mehr korrekt funktionieren wird! Für den Abgleich wird ein zweiter, separater Schlüsselwert von CMI LehrerOffice zur Verfügung gestellt, der frei an die Schulverwaltung angepasst werden darf.

subjectNiveau optional

Niveauangabe bei einem Niveaufach.

Format: [alphanumerisch](#)

Beispiel: [B](#)

subjectAddition optional

Zusatzangabe zum Fach (wie Instrument bei Musikunterricht oder Sprache bei HKS).

Format: [alphanumerisch](#)

Beispiel: [Trompete](#)

<code><groupData></code>	<i>Stammdaten der Arbeitsgruppe</i>
--------------------------------	-------------------------------------

lehreroffice/inst/semesters/semester/studyGroups/studyGroup/groupData

5.3.8 Element 'groupData'

empfohlen

Im Gegensatz zum Element 'groupDefinition' werden in diesem Element die semesterabhängigen Grunddaten der aktuellen Schulklasse überliefert.

Anzahl: 1

Beispiel:

```
<groupData>
  ...
</groupData>
```

5.3.9 Elemente zu 'groupData'

schoolhouse

empfohlen

Bezeichnung des Schulhauses. Diese Information wird innerhalb CMI LehrerOffice semesterabhängig abgelegt.

Format: `alphanumerisch`

Beispiel: `Wisli`

schoolhouseKey

empfohlen

Eindeutige ID vom Quellsystem für das Schulhaus. Die Schulhäuser müssen vorab innerhalb von CMI LehrerOffice definiert und mit demselben Schlüsselwert wie im Quellsystem eingestellt werden.

Format: `alphanumerisch`

Beispiel: `WSI`

room

optional

Nummer des Schulzimmers / Klassenzimmers.

Format: `alphanumerisch`

Beispiel: `A21`

6 Daten einer Arbeitsgruppe (Fachklasse / Niveaunklasse)

Mit dieser Methode können die Zuteilungen der Schüler- und Lehrpersonen einer manuellen Arbeitsgruppe (Fachklasse, Niveaunklasse) übernommen werden. Diese Angaben werden schlussendlich im Modul 'Gruppen' von CMI LehrerOffice abgelegt werden.

Daten einer Arbeitsgruppe

- Schülerzuordnung
- Lehrpersonenzuteilungen

SOAP-Methode: `IO_getStudyGroup`
Parameter: `i` `nt semesterYear, int semesterPart, String groupKey`

Alternative SOAP-Methode: `IO_getStudyGroup2`
Parameter: `int semesterYear, int semesterPart, String groupKey, String groupID`

Dateiname: `"SJ" + Schuljahr + "-" + Semester + "_Gruppe_" + Gruppenkürzel + ".xml"`

Beispiel: `SJ2011-2_Gruppe_EnglischE1.xml`

Lehrpersonen und die Schülerdaten müssen vorgängig importiert sein.

6.1 SOAP – Anfrage für Daten einer Arbeitsgruppe

<LO_getStudyGroup>

Methodenaufruf mit Parameter

soap:Envelope/soap:Body/LO_getStudyGroup

6.1.1 Element 'LO_getStudyGroup'

zwingend notwendig

Dieses Element ist der eigentliche Methodenaufruf für die Abfrage/Aktion der Gruppendaten mit den dazugehörigen Schülern.

Jede Schulverwaltungslösung kann diese SOAP-Methode als Web-Service zur Verfügung stellen. Das erwartete Resultat als XML ist im nachfolgenden Abschnitt dokumentiert.

Beispiel:

```
<soap:Body>
  <LO_getStudyGroup xmlns="http://www.lehrer-
office.ch/xmlns/soap/2010/exchange-v1">
    <semesterYear>2010</semesterYear>
    <semesterPart>1</semesterPart>
    <groupKey>134</groupKey>
  </LO_getStudyGroup>
</soap:Body>
```

6.1.2 Attribute zum Element 'LO_getStudyGroup'

xmlns="http://www.lehreroffice.ch/xmlns/soap/2010/exchange-v1"

zwingend notwendig

Dieser Namespace muss als Standardnamespace, also ohne Namespace-Präfix angegeben werden.

Diese Versionsangabe hat keinen Zusammenhang mit der Programmversion von CMI LehrerOffice als Anwendung.

Format: URL

Beispiel: <http://www.lehreroffice.ch/xmlns/soap/2010/exchange-v1>

6.1.3 Elemente zu 'LO_getStudyGroup' als Parameter

`semesterYear` zwingend notwendig

Vierstellige Ganzzahl für das Schuljahr, wobei das Kalenderjahr des Schulanfangs geliefert wird. Für das Schuljahr 2010/11 wird somit die Zahl '2010' geliefert.

Format: `numerisch (4)`

Beispiel: `2010`

`semesterPart` zwingend notwendig

Einstellige Ganzzahl für die Semesterangabe.

Format: `1|2`

Beispiel: `1`

`groupKey` zwingend notwendig

Eindeutiger Schlüssel für die Identifikation der Arbeitsgruppe resp. Fachklasse.

Format: `alphanumerisch`

Beispiel: `134`

`groupID` optional

Eindeutige ID bei einem Kurssystem mit laufender, schuljahresabhängiger ID (nur für EcoOpen und Sclaris). Dieser Parameter wird nur geliefert, wenn in CMI LehrerOffice eine laufende ID erfasst wurde.

Format: `alphanumerisch`

Beispiel: `1004`

6.2 Daten einer Arbeitsgruppe – XML-Struktur

XML-Tag	Dateninhalt
<lehreroffice>	Dateninhalt
<inst>	Institution / Schulgemeinde
<source>	Angaben zum Quellsystem
</source>	
<semesters>	Auflistung der Semester
<semester>	Semesterdaten
<studygroups>	Auflistung der Arbeitsgruppen
<studygroup>	Arbeitsgruppe
<groupDefinition>	Gruppendefinition
</groupDefinition>	
<groupData>	Stammdaten der Arbeitsgruppe
</groupData>	
<groupTeachers>	Auflistung aller Lehrpersonen dieser Gruppe
<groupTeacher>	Lehrperson
</groupTeacher>	
</groupTeachers>	
<students>	Auflistung aller Schüler/innen
<student>	Schüler/in
</student>	
</students>	
</studygroup>	
</studygroups>	
</semester>	
</semesters>	
</inst>	
</lehreroffice>	

6.3 Daten einer Arbeitsgruppe – Elemente

<code><lehreroffice></code>	<i>Dateninhalt</i>
<code><inst></code>	<i>Institution / Schulgemeinde</i>
<code><source></code>	<i>Angaben zum Quellsystem</i>
<code></source></code>	
<code><semesters></code>	<i>Auflistung der Semester</i>
<code><semester></code>	<i>Semesterdaten</i>
<code></semester></code>	
<code></semesters></code>	
<code></inst></code>	
<code></lehreroffice></code>	

Obige Elemente und dessen Attribute und Unterelemente sind im vorangehenden Kapitel ' Daten für Lehrpersonen – Elemente' detailliert dokumentiert. Nachfolgend sind nun die Elemente innerhalb vom Element '<semester>' beschrieben.

`<studyGroups>`

Auflistung der Gruppen

lehreroffice/inst/semesters/semester/studyGroups

6.3.1 Element 'studyGroups'

zwingend notwendig

Dieses Element fungiert als Liste mit Unterelementen vom Typ '`<studyGroup>`'. In der aktuellen Version wird nur eine einzelne Gruppe unterstützt. In künftigen Versionen sind unter Umständen mehrere Gruppen innerhalb einer XML-Datei möglich.

Anzahl: 1

Beispiel:

```
<studyGroups complete="0">
  <studyGroup>
    ...
  </studyGroup>
</studyGroups>
```

<studyGroup>

Arbeitsgruppe

lehreroffice/inst/semesters/semester/studyGroups/studyGroup

6.3.2 Element 'studyGroup'

zwingend notwendig

Datenelement einer Arbeitsgruppe (Niveaunklasse, Fachklasse). Es wird empfohlen pro Gruppe eine XML-Datei zu exportieren. Dies vereinfacht die allfällige Fehlersuche das wiederholte Einlesen von Gruppendaten.

Anzahl: 1..n

Beispiel:

```
<studyGroup key="SHA-E2" id="19" modDateTime="2010-01-03T09:32:59">  
  ...  
</studyGroup>
```

6.3.3 Attribute zum Element 'studyGroup'

key="SHA-E2"

zwingend notwendig

Eindeutige und statische ID vom Quellsystem für die Arbeitsgruppe, welche über alle Schuljahre unverändert bleibt.

Format: alphanumerisch

Beispiel: SHA-E2

In CMI LehrerOffice werden die Gruppengefäße einmalig definiert, wobei dazu für jede Arbeitsgruppe das dazugehörige Schulfach und Niveau hinterlegt wird. Die einzelnen Schüler und die entsprechenden Lehrpersonen werden am Ende eines Schuljahres nun von der aktuellen Gruppeneffinition zur nächsten weitergeschoben. Vergleichbar mit einem Klassenzimmer, das immer wieder mit neuen Schülern und Lehrpersonen belegt wird.

id

optional

Bei Schulverwaltungslösungen (wie EcoOpen, Sclaris), welche für jede Schulklasse resp. Gruppe eine neue ID zu Beginn der Schulzeit generieren, kann die laufende ID in CMI LehrerOffice auch als zweites Suchkriterium genutzt werden.

Schulverwaltungslösungen (wie iCampus), welche dasselbe System mit den fixen Schlüsselwerten wie CMI LehrerOffice anwendet, darf dieses Attribut nicht nutzen.

Format: alphanumerisch

Beispiel: 19

modDateTime

optional

Modifikationsdatum im XSD-Format. Letzte Mutation des Datensatzes auf dem Quellsystem. Wenn dieser Zeitstempel älter oder gleich dem abgespeicherten TimeStamp in CMI LehrerOffice ist, wird dieses Element und alle Unter-elemente nicht verarbeitet. Dank diesem TimeStamp kann die Verarbeitungszeit stark optimiert werden.

Auch bei einer Änderung der dazugehörigen Lehrpersonen, der untergeordneten Schülerzuteilungen muss ein geänderter Zeitstempel geliefert werden!

Format: YYYY-MM-DDTHH:MM:SS

Beispiel: 2010-01-03T09:32:59

active

optional

Arbeitsgruppe ist in diesem Semester aktiv (1) gekennzeichnet oder nicht (0).

Format: `numerisch 0|1 (1)`

Beispiel: `1`

<groupDefinition>

Gruppendefinition

lehreroffice/inst/semesters/semester/studyGroups/studyGroup/groupDefinition

6.3.4 Element 'groupDefinition'

(nicht implementiert) optional

Damit künftig Gruppengefässe automatisch in CMI LehrerOffice erstellt werden können, kann die Definition der Arbeitsgruppe mit diesen Angaben einmalig erstellt werden, wenn die Gruppe gemäss Schlüssel nicht gefunden werden konnte. Diese Definitionen sind im Gegensatz zu 'groupData' nicht semesterabhängig.

Anzahl: 1

Beispiel:

```
<groupDefinition>
  ...
</groupDefinition>
```

6.3.5 Attribute zu 'groupDefinition'

modDateTime optional

Modifikationsdatum im XSD-Format. Letzte Mutation des Datensatzes auf dem Quellsystem. Wenn dieser Zeitstempel älter oder gleich dem abgespeicherten TimeStamp in CMI LehrerOffice ist, wird dieses Element und alle Unterelemente nicht verarbeitet. Dank diesem TimeStamp kann die Verarbeitungszeit stark optimiert werden.

Dieser Zeitstempel gilt nur für die Gruppendefinition und nicht für die folgenden Schülerzuteilungen.

Format: YYYY-MM-DDTHH:MM:SS

Beispiel: 2010-01-03T09:32:59

6.3.6 Elemente zu 'groupDefinition'

name optional

Allgemeingültige Gruppenbezeichnung als Kürzel, wobei semesterabhängige Angaben und Namen von Lehrpersonen zu unterlassen sind.

Format: alphanumerisch

Beispiel: Englisch II

unitKey optional

Grosse Schulgemeinden können in Schuleinheiten unterteilt werden, wobei jede Arbeitsgruppe einer Schuleinheit zugeordnet wird.

Eindeutige ID vom Quellsystem für die Schuleinheit. Die Schuleinheiten müssen vorab innerhalb von CMI LehrerOffice definiert und mit demselben Schlüsselwert wie im Quellsystem eingestellt werden.

Format: alphanumerisch

Beispiel: OZ1

unitName optional

Name der Schuleinheit. Dient lediglich als Information für die Zuordnung der Schuleinheit zur ID, wird nicht übernommen.

Format: [alphanumerisch](#)

Beispiel: [Sekundarschule](#)

unitCity optional

Schulort der Schuleinheit.

Format: [alphanumerisch](#)

Beispiel: [Thayngen](#)

bur="72765790" optional

Zur Identifikation wird die BUR-Nummer SSP vom Amt für Statistik verwendet. Diese BUR-Nummer SSP ist ein eindeutiger Identifikator der Bildungsinstitute der Schweiz.

Format: [numerisch \(20\)](#)

Beispiel: [72765790](#)

subjectKey optional

Eindeutige und statische ID vom Quellsystem für das Schulfach, welche über alle Schuljahre unverändert bleibt. Wenn das Schulfach in CMI LehrerOffice und diese Angabe nicht übereinstimmt, wird eine Warnung ausgegeben.

Format: [alphanumerisch](#)

Beispiel: [D](#)

In CMI LehrerOffice werden die Schulfächer einmalig definiert und einmalig mit dem dazugehörigen Schlüssel der Schulverwaltung ergänzt.

Bei den Fachdefinitionen darf der Wert im Feld 'Schlüssel' von CMI LehrerOffice nicht verändert werden, da sonst der Zeugnisdruck nicht mehr korrekt funktionieren wird! Für den Abgleich wird ein zweiter, separater Schlüsselwert von CMI LehrerOffice zur Verfügung gestellt, der frei an die Schulverwaltung angepasst werden darf.

subjectNiveau optional

Niveauangabe bei einem Niveaufach.

Format: [alphanumerisch](#)

Beispiel: [B](#)

subjectAddition optional

Zusatzangabe zum Fach (wie Instrument bei Musikunterricht oder Sprache bei HKS).

Format: [alphanumerisch](#)

Beispiel: [Trompete](#)

<groupData>

Stammdaten der Arbeitsgruppe

lehreroffice/inst/semesters/semester/studyGroups/studyGroup/groupData

6.3.7 Element 'groupData'

empfohlen

Im Gegensatz zum Element 'groupDefinition' werden in diesem Element die semesterabhängigen Grunddaten der aktuellen Schulklasse überliefert.

Anzahl: 1

Beispiel:

```
<groupData>  
  ...  
</groupData>
```

6.3.8 Elemente zu 'groupData'

schoolhouse

empfohlen

Bezeichnung des Schulhauses. Diese Information wird innerhalb CMI LehrerOffice semesterabhängig abgelegt.

Format: [alphanumerisch](#)

Beispiel: [Wisli](#)

schoolhouseKey

empfohlen

Eindeutige ID vom Quellsystem für das Schulhaus. Die Schulhäuser müssen vorab innerhalb von CMI LehrerOffice definiert und mit demselben Schlüsselwert wie im Quellsystem eingestellt werden.

Format: [alphanumerisch](#)

Beispiel: [WSI](#)

schoolhouseCity

optional

Schulort des Schulhauses.

Format: [alphanumerisch](#)

Beispiel: [Thayngen](#)

room

optional

Nummer des Schulzimmers / Klassenzimmers.

Format: [alphanumerisch](#)

Beispiel: [A21](#)

<code><groupTeachers></code>	<i>Auflistung aller Lehrpersonen</i>
------------------------------------	--------------------------------------

lehreroffice/inst/semesters/semester/studyGroups/studyGroup/groupTeachers

6.3.9 Element 'groupTeachers'

empfohlen

Dieses Element fungiert als Liste mit den entsprechenden Adresselementen vom Typ '<groupTeacher>'. Diese Liste umfasst alle Lehrpersonen und Heilpädagogen dieser Klasse.

Die Lehrpersonen müssen vorab als Adressen innerhalb von CMI LehrerOffice vorhanden sein. Aus diesem Grund werden die Lehrpersonenadressen zuerst importiert, damit die aktuellen Adressdaten der Lehrpersonen zur Verfügung stehen.

Anzahl: 1

Beispiel:

```
<groupTeachers complete="1">
  <groupTeacher ahv="756.0978.6543.21" key="1001" function="HL" />
  <groupTeacher key="1323" function="WL" />
</groupTeachers>
```

6.3.10 Attribute zum Element 'groupTeachers'

complete="1"

zwingend notwendig

Wenn der Code "1" (Standard) übergeben wird, dann ist die nachfolgende Liste mit den Lehrpersonen vollständig. Dies bedeutet, dass alle Lehrpersonen enthalten sind. Überzählige Lehrpersonen werden hiermit automatisch innerhalb CMI LehrerOffice von dieser Gruppe entfernt.

Mit dem Code "0" werden die fehlenden Lehrpersonen hinzugefügt. Diese Option ist für das Nachliefern von einzelnen Lehrpersonen sinnvoll (Push-Variante).

- 0 = Liste mit Lehrpersonen ist nicht vollständig
- 1 = Liste mit Lehrpersonen ist komplett (Standard)

Format: numerisch 0|1 (1)

Beispiel: 1

<code><groupTeacher></code>	<i>Lehrperson</i>
-----------------------------------	-------------------

lehreroffice/inst/semesters/semester/studyGroups/studyGroup/groupTeachers/groupTeacher

6.3.11 Element 'groupTeacher'

empfohlen

Datenelement für das Zuordnen einer Lehrperson. Dieses Element enthält keine weiteren Elemente, sondern dient lediglich für die Zuteilung der Lehrpersonen, welche im Adressstamm bereits enthalten sein müssen.

Anzahl: 0..3

Beispiel:

```
<groupTeacher ahv="756.0978.6543.21" key="1001" function="HL" />
```

6.3.12 Attribute zum Element 'groupTeacher'

ahv="756.0978.6543.21"

empfohlen

AHV-Nummer dient als Identifikator mit erster Priorität. Die Datenschnittstelle von CMI LehrerOffice sucht zuerst nach einem Adressen-Datensatz mit entsprechender AHV-Nummer. Wenn erfolgreich, wird diese Adresse als Klassenlehrperson zugeteilt. Sollte der Datensatz nicht gefunden werden, wird in zweiter Priorität nach dem Attribut 'key' gesucht (siehe nachfolgende Beschreibung).

Format: [alphanumerisch \(16\)](#)

Beispiel: 756.0978.6543.21

key="1001"

zwingend notwendig

Eindeutige ID vom Quellsystem für die entsprechende Adresse der Lehrperson.

Wenn die Schulverwaltung noch keine gültigen AHV-Nummern besitzt kann in zweiter Priorität mit diesem Schlüsselwert abgeglichen werden. Sobald eine AHV-Nummer vorhanden ist, wird zuerst nach dieser gesucht.

Format: [alphanumerisch](#)

Beispiel: 1001

function="HL"

optional

Funktion resp. Rolle dieser Lehrperson

Werte: [HL](#) = Lehrperson als verantwortliche Hauptlehrperson (max. 1 LP)

[WL](#) = Weitere Lehrperson bei Jobsharing (max. 2 LP)

Beispiel: [HL](#)

`<students>` *Auflistung aller Schüler/innen*

lehreroffice/inst/semesters/semester/studyGroups/studyGroup/students

6.3.13 Element 'students'

zwingend notwendig

Dieses Element fungiert als Liste mit den entsprechenden Elementen vom Typ '<student>'. Diese Liste umfasst die Schüler/innen und dessen Niveau- und Wahlfachzuteilungen der entsprechenden Gruppe.

Dies bedeutet, wenn ein Schüler einem Wahlfach noch nicht zugeteilt ist und die Gruppe ein Wahlfach betrifft, wird der Schüler dem Wahlfach automatisch hinzugefügt. Wenn der Schüler aus der Gruppe entfernt und nicht vorgängig von einer anderen Gruppe bereits zugewiesen wurde, dann wird das Wahlfach entsprechend entfernt.

Anzahl: 1

Beispiel:

```
<students complete="1">  
  ...  
</students>
```

6.3.14 Attribute zum Element 'students'

`complete="1"`

zwingend notwendig

Wenn der Code "1" (Standard) übergeben wird, dann ist die nachfolgende Schülerliste vollständig. Dies bedeutet, dass alle Schüler von entsprechender Arbeitsgruppe enthalten sind. Überzählige Schüler innerhalb von CMI Lehrer-Office werden hiermit automatisch aus der Gruppe entfernt.

Mit dem Code "0" werden die fehlenden Schüler hinzugefügt. Diese Option ist für das Nachliefern von einzelnen Schülern sinnvoll (Push-Variante).

- 0 = Schülerliste ist nicht vollständig
- 1 = Schülerliste ist komplett (Standard)

Format: `numerisch 0|1 (1)`

Beispiel: 1

<code><student></code>	<i>Schüler/in</i>
------------------------------	-------------------

lehreroffice/inst/semesters/semester/studyGroups/studyGroup/students/student

6.3.15 Element 'student'

wenn Daten vorhanden

Datenelement für das Zuordnen eines Schülers. Dieses Element enthält keine weiteren Elemente, sondern dient lediglich für die Zuteilung des Schülers, welcher bereits im Modul 'Personaldata' von CMI LehrerOffice enthalten sein müssen.

Anzahl: 0..n

Beispiel:

```
<student ahv="456.1941.1243.92" key="9303" caption="Anna Meier"/>
```

6.3.16 Attribute zum Element 'student'

ahv="456.1941.1243.92"

empfohlen

AHV-Nummer des Schülers dient als Identifikator mit erster Priorität. Die Datenschnittstelle von CMI LehrerOffice sucht zuerst nach einem gültigen Schüler-Datensatz mit entsprechender AHV-Nummer. Sollte der Datensatz nicht gefunden werden, wird in zweiter Priorität nach dem Attribut 'key' gesucht (siehe nachfolgende Beschreibung).

Format: alphanumerisch (16)

Beispiel: 456.1941.1243.92

key="9303"

zwingend notwendig

Eindeutige ID vom Quellsystem für den Schüler.

Wenn die Schulverwaltung noch keine gültigen AHV-Nummern besitzt kann in zweiter Priorität mit diesem Schlüsselwert abgeglichen werden. Sobald eine AHV-Nummer vorhanden ist, wird zuerst nach dieser gesucht.

Format: alphanumerisch

Beispiel: 9303

caption=""

empfohlen

Für die Fehlersuche kann ein Text mit dem Namen des Schülers mitgeliefert werden. Diese Angabe wird aber nicht weitergehend verarbeitet und dient lediglich für eine allfällige Fehlersuche und Protokollausgabe.

7 Anhang

7.1 XML-Entities

Im XML-Format gibt es vordefinierte Sonderzeichen, die für die Steuerung des XML-Prozessors zuständig sind und daher nicht als Textinhalt benutzt werden dürfen. Falls man dies nicht beachtet, gilt das XML-Dokument als nicht wohlgeformt und kann nicht verarbeitet werden.

Falsch: `<name>Herr Von & Zu S'antonella ("Sani")</name>`
`<formel>1 < 2</formel>`

Richtig: `<name>Herr Von & Zu S'antonella ("Sani")</name>`
`<formel>1 < 2</formel>`

Folgende Sonderzeichen müssen durch ihre XML-Entities ersetzt werden:

Name	Zeichen	Unicode	XML-Entity
Doppelte Anführungszeichen (Quotation Mark)	"	0022	"
Einfache Anführungszeichen (Apostrophe)	'	0027	'
Kaufmännisches Und (Ampersand)	&	0026	&
Kleiner-Als Zeichen (Less-Than Sign)	<	003C	<
Grösser-Als Zeichen (Greater-Than Sign)	>	003E	>

Tipp: Falls wie im ersten Beispiel mehrere Ersetzungen notwendig sind, sollte man immer zuerst das Und-Zeichen („&“) ersetzen, da sonst die anderen Entities davon beeinflusst werden.

7.2 Rechtliche Hinweise und Urheberrechte

Die vorliegende XML-Struktur wurde durch die Firma Roth Soft erstellt. Somit liegen sämtliche Urheberrechte bei der Firma Roth Soft. Roth Soft ist berechtigt, jederzeit Aktualisierungen und Erweiterungen der XML-Definitionen nach eigenem Ermessen durchzuführen oder auch die Datenschnittstelle auf Seiten von CMI LehrerOffice abzuändern.

Diese XML-Struktur darf nur im Zusammenhang mit dem Datenaustausch zu CMI LehrerOffice verwendet werden. Jegliche anderweitige Nutzung, Weiterentwicklung oder Weiterverbreitung als Ganzes oder in Teilen bedarf der schriftlichen Zustimmung von Roth Soft. Roth Soft behält sich das Recht vor, die Nutzung dieser Schnittstelle einzelner Anbieter von Schulverwaltungslösungen zu verbieten.

Der Gebrauch von bereitgestellter Software oder von Web-Services erfolgt auf eigenes Risiko der Anwender, welche auch die notwendigen Vorkehrungen zur Datensicherung sowie für den Datenschutz zu treffen haben. Roth Soft ist nicht für irgendwelche Schäden (wie Betriebsunterbrechungen, Verluste von Informationen, finanzielle Verluste usw.) ersatzpflichtig. Es ist untersagt, die Software von Roth Soft abzuändern, zu übersetzen, zurückzuentwickeln, zu dekompileieren oder abgeleitete Werke zu erstellen.

Der Produktname "CMI LehrerOffice" ist ein eingetragenes Markenzeichen von Roth Soft. Als Gerichtsstand wird Andelfingen genannt. Mit dem Anwenden der XML-Definitionen von Roth Soft werden die vorstehend beschriebenen Bedingungen vollumfänglich anerkannt.