

Vigilo OneRoster implementation

Table of content

1	Change log	2
2	Required	4
3	Restrictions (might change)	4
4	Cache	5
5	Endpoints	6
6	Additional Metadata	9
6.1	Overview of GroupTypes in Vigilo OAS	13
7	Implementation guide	15
7.1	How to exclude previous administrative staff in school	16
7.2	Implementation notes	17
8	Endpoints - Childcare extension	18
9	Endpoints – AfterSchool extension	19
10	Endpoints – Application extension	20
11	Endpoints – Sessions	22
12	Endpoints – Student absence	25

1 Change log

- 09.06.2022** Chapter 6: ORG: Metadata-field `externalOrganizationNumber` marked as deprecated, and replaced by `hrUnitIds` (array of strings)
Chapter 6: USER: Extended metadata (`previousOrg` for `role=Student`)
- 19.05.2022** Chapter 6: Extended metadata for USER.
Chapter 7: Step 7. Included “`employeeStatus`” as option for finding administrative staff in school
Chapter 7.1: Changed implementation note. Old note is deprecated.
Chapter 7.2: Rephrased implementation note for `startDate / endDate` definitions
Chapter 7.2: Childcare employee: Deprecated implementation note for childcare employee position info.
Chapter 11: Pre-warning that data type for ***week number*** might be changed from text to number in future release.
- 25.01.2022** Chapter 6: Marked ORG metatags `nationalRegisterId` and `gsild` as “deprecated”
Changed description for ORG metatag `shortName` from “Organization pasCode” to “Organization short-name”.
- 17.01.2022** Chapter 6: Added `userIdsMerged` as metadata-tag on USER to identify the IDs of merged persons from Vigilo’s “duplicate handling process”.
Chapter 6: Marked metatag `nationalRegisterId` on ORG as “to be deprecated”
- 13.01.2022** Chapter 5: New endpoint “GetEnrollmentsForStudent” added in list
Chapter 7.2: Minor clarifications regarding employment data for childcare employees in new chapter 7.2.
Chapter 7.2: Changed behavior for “`endDate`” for `academicSessions/terms` in new chapter 7.2
New chapter 11: Endpoints - Sessions
New chapter 12: Endpoints - Student absence
- 14.12.2021** Due to structural changes in Vigilo, some changes will occur on teachers; Deprecated position information is now gone. More position-metadata tags are available; “`employmentType`” and “`positionPercentage`”. Status will come with different case than

before (e.g “Active” will now come as “active”) – make sure you treat it case-insensitive! Performance on /users and /teachers may be a bit better.

- 10.11.2021** New chapter 10: Endpoints – Application extension

- 28.10.2021** Chapter 3 “Restrictions”: Included a general disclaimer regarding release of undocumented changes.
Chapter 6 “Additional metadata”: Included sub-chapter 6.1 with description of GroupTypes used in Vigilo OAS (partly in Norwegian).
New chapter 9: Endpoints – AfterSchool extension

- 30.09.2021** Extended implementation guide with sub-chapter: “How to exclude previous administrative staff in school”

- 31.08.2021** Added documentation about childcare-extension. Deprecated position-information will be removed from the API shortly. Added chapter about the cache-mechanism.

- 24.02.2021:** Metadata-tag “ownership” added to orgs(/schools/childcares). Added classType in filters available for enrollments. Notice about classType filter in “Implementation guide”.

- 03.02.2021:** Fixed typo in getClassesForStudent URI and added restriction-note about parallelism.

- 17.11.2020:** Note about OneRoster 1.1 compliance. Added more metadata onto user; primaryOrg, schoolTransport*. Removed doc about deprecated position-information for teachers.

2 Required

TenantId

IP of caller (for ip-filter pr Tenant).

Use HTTP-header for authentication: Key: <tenantId>, value: <token given by Vigilo>

3 Restrictions (might change)

- Due to performance, some of the most resource-heavy resources have been disabled, and will return “403 Forbidden”. In that case, use /schools/<schoolId>/xxx endpoint instead.
- dateLastModified will always be todays date at midnight
- Some SourceIds contains a composite key. They will contain ‘\’ to separate ids and it will be urlEncoded.
- Filters will be added by request/need.

We HIGHLY recommend using the resources pr. organization. (/schools/) instead of root as most of the data makes more sense in context of an organization. Most of the data is organization specific and are not to be used across several organizations.

Don't use parallel threads while retrieving data as this might overload the system and will not take advantage of built-in cache mechanisms and make the whole API perform slower.

Example of pagination: /students?limit=50&offset=250

Example of filter: school/<schoolId>/students?filter=status='active'

Example of both; school/<schoolId>/students?filter=status='active'&limit=50&offset=250

Note: Vigilo OAS is now **OneRoster 1.1 API certified** partner. However, current production endpoints are **not** the certified ones, as there were quite a few changes that had to be made in order to become compliant with 1.1. (and all you consumers would have to make some changes in order to make it work). We are currently working on the **1.2** version of the OneRoster implementation, and will publish this when ready as it seems to be closer to the needs of the Norwegian school-model. That's why we will wait with the certified product until ready on 1.2. - and keep you from having to change your clients more than needed.

General disclaimer:

Vigilo might at any given time release API-changes that:

- Add metadata to existing API endpoints
- Add new API endpoints (for beta-testing before official release).

Users of the API should consume payload in a way that handle new data and new endpoints in a way that it does cause errors on the consumer side.

4 Cache

Due to large amounts of data, some entities are cached in order to perform better and provide less load on the Vigilo suite. Request should be performed in serial and not parallel due to getting the desired performance boost when retrieving data which is cached. Due to cache-implementation, some data changes in Vigilo might not show in the API for some time;

Entity	Time for cache to invalidate (up to)
Users (/students/teachers/parents)	18 hrs
Organizational unit (/school/childcare)	4 hrs
SchoolYears and terms	4 hrs
Subjects (used in e.g classes)	8 hrs

5 Endpoints

Base url: <https://dataporten-api.vigilo.no:4004/api/<tenantId>ims/oneroster/v1p1>

Action	URI	Supported filter *
getAllOrgs	/orgs	
getOrg	/orgs/<orgId>	
getAllSchools	/schools	
getSchool	/schools/<schoolId>	
getAllAcademicSessions	/academicSessions (restricted due to heavy load)	type: schoolYear term status: active tobedeleted
getAcademicSessionsForSchool	/schools/<schoolId>/academicSessions	type: schoolYear term status: active tobedeleted
getAcademicSessionsForOrg	/orgs/<orgId>/academicSessions (not for childcares)	type: schoolYear term status: active tobedeleted
getAcademicSession	/academicSessions/<academicSessionId> Alt: /schools/<schoolId>/academicSessions/<academicSessionId> /orgs/<orgId>/academicSessions/<academicSessionId>	
getAllUsers	/users	Identifier role: parent
getUsersForSchool	/schools/<schoolId>/users	Identifier status: active tobedeleted
getUser	/users/<userId>	

Action	URI	Supported filter *
	Alt: /schools/<schoolId>/users/userId	
getAllStudents	/students	identifier
getStudentsForSchool	/schools/<schoolId>/students	identifier status: active tobedeleted
getStudentsForClassInSchool	/schools/{school_id}/classes/{class_id}/students	Identifier status: active tobedeleted
getStudent	/students/<userId> Alt: /schools/<schoolId>/students/<userId>	
getAllTeachers	/teachers	identifier
getTeachersForSchool	/schools/<schoolId>/teachers	identifier
getTeachersForClassInSchool	/schools/{school_id}/classes/{class_id}/teachers	identifier
getTeacher	/teachers/<userId> Alt: /schools/<schoolId>/teachers/<userId>	
getAllClasses	/classes (restricted due to heavy load)	schoolYearId: <schoolYearId> classType: homeroom scheduled - OR - groupType: Vigilo-internal value which is presented in metadata tag
getClassesForSchool	/schools/<schoolId>/classes/	status: N/A active tobedeleted schoolYearId: <schoolYearId> classType: homeroom scheduled - OR - groupType: Vigilo-internal value which is presented in metadata tag
getClassesForStudent	/school/<schoolId>/students/<studentId>/classes	schoolYearId: <schoolYearId>

Action	URI	Supported filter *
		classType: homeroom scheduled - OR - groupType: Vigilo-internal value which is presented in metadata tag
getClass	/classes/<classId> Alt: /schools/<schoolId>/classes/<classId>	
getAllCourses	/courses (restricted due to heavy load) In development	
getCoursesForSchool	/schools/<schoolId>/courses/ In development	
getCourse	In development	
getAllEnrollments	/enrollments (restricted due to heavy load unless schoolYearId is specified; /enrollments)	schoolYearId: <schoolYearId> status: active tobedeleted role: teacher student
GetEnrollmentsForStudent	/school/<schoolId>/students/<studentId>/enrollments	schoolYearId: <schoolYearId> status: active tobedeleted
getEnrollmentsForSchool	/schools/<schoolId>/enrollments	schoolYearId: <schoolYearId> status: active tobedeleted role: teacher student classType: homeroom scheduled
getEnrollmentsForClassInSchool	/schools/{school_id}/classes/{class_id}/enrollments	status: active tobedeleted role: teacher student
getEnrollment	/enrollments/<enrollmentId> /schools/<schoolId>/enrollments/<enrollmentId>	
getTerms	/terms	
getTermsForSchool	/schools/<schoolId>/terms	

Action	URI	Supported filter *
getTerm	/terms/<termId> Alt: /schools/<schoolId>/terms/<termId>	

6 Additional Metadata

The “metadata”-tag of the payloads consist of additional data which can be useful. It’s basically a key-value map which does not have any specific parameters according to the standard but are populated on-demand with data which can be useful for the consumer of the API. The table below lists some of the values currently available. Please note - when there’s an empty value, the key might also not come.

Payload	Metadata
Org	<p>academicSessions: Link to AcademicSessions for this org. classes: Link to Classes for this org. students: Link to Students (user-objects) for this org. teachers: Link to Teachers (user-objects) for this org. enrollments: Link to Enrollments for this org. courses: Link to Courses for this org. externalOrganizationNumber: OrganizationUnit – externalOrganizationNumber (deprecated, and will be removed without further notice, as it is replaced by hrUnitIDs. Note that multiple values are divided by COMMA, e.g. 35211, 35240). addressline1: Organization addressline1 addressline2: Organization addressline2 postalCode: Organization postalCode city: Organization city latitude: Organization latitude longitude: Organization longitude phoneNumber: Organization phoneNumber email: Organization email homePageUrl: Organization homePageUrl managerFirstName: Organization managerFirstName</p>

Payload	Metadata
	<p>managerLastName: Organization managerLastName vigold: Organization vigold shortName: Organization short-name key: Only for tenants: Tenant “key”. ownership: Schools: “Privat” ”Offentlig” ChildCares: “Privat” ”Kommunal” hrUnitIDs: (new): Array of strings from user interface «Enhetsinformasjon/ Tilleggsinformasjon», e.g. "hrUnitIds": ["35211","35240"]</p>
<p>User Note: 1 person might be several users with the same sourceId, but with different roles and orgs. (student/teacher/parent). 1 user might also come as several students or teachers - one pr org where they belong. E.g: 1 user is a parent + 50% teacher at two schools + taking a course at a third school = 1 parent + 2 teachers + 1 student occurrence.</p>	<p>formOfWrittenNorwegian: nn nb firstLanguage: Language codes according to ISO 639 alpha 3 gender: male female. birthDate: Date of birth (format: yyyy-MM-dd). fromDate: (student) fromDate (Not currently in use!) toDate: (student) toDate (Not currently in use!) userIdsMerged: The IDs of persons that has been merged into this (previous persons has been removed) as a part of duplicate handling process in Vigilo previousOrg: (new) SourcedId for a student's <u>previous school</u> within same tanant (e.g. “Avgiverskole” as sent to “PAS Prøver API”). Only returned if existing, and only for a single student API request.</p> <p>Only returned when getting in the context of an organization: primaryOrg: true false – For students (not in use for childcare) this is called “hospiteringselev” in user interface. For employees this is called “primærenhet” in user interface. employeeNumber: Employee number employeeFromDate: Start-date for the employee employeeToDate: End-data (until, not included) for the employee employeeStatus: (active, starting, ended): The status calculated by the from-date/to-date interval inactiveDate: (student) inactiveDate schoolTransportFromDate: (student) FromDate this student is eligible for free school transport schoolTransportEndDate: (student) EndDate this student is eligible for free school transport schoolTransportDescription: (student) School transport description.</p> <p>positions: (teacher) Array of positions for the teacher (even for childcare employees). May contain the following data:</p>

Payload	Metadata
	<p> positionName: (teacher) positionName positionDescription: (teacher) positionDescription positionPercentage: (teacher) positionPercentage. employmentType: (teacher) employmentType (see positionType – will be deprecated) positionType: (teacher) positionType status: (teacher) employee position status (starting, quit, active). startDate: position start date endDate: position end date (until, not included) </p> <p> studentId: N/A: internal use only. clientId: N/A: internal use only. personId: N/A: internal use only (same as sourcedId). employeeId: N/A: internal use only </p> <p> userIds (not in metadata-tag, but documented here as it's dynamic): adUserIdentifier: ActiveDirectory username. feideUserIdentifier: Feide username. upnUserIdentifier: UPN username. employeeNumber: Employee number </p>
AcademicSession	<p> status: status of the schoolYear directly from Vigilo. "Nåværende Utgått Pre_registered" org: Link to the Organization of which this academicSession belongs. </p>
Class	<p> schoolYear: Link to the academicSession (type: schoolYear) this class belongs to. pasPeriod: If this instance is a PAS-group, this will display PAS Period. pasSubjectCode: If this instance is a PAS-group, this will display PAS SubjectCode. groupType: GroupType from Vigilo. Possible values to date: (Class, LearningSupport, SubjectGroup, SnoGroup, GnoGroup, SfoGroup, SfaGroup, TpoGroup, NativeLanguage, TofGroup, OtherUsers, PasTestGroup) pasTransferRequestedByPersonId: If this instance is a PAS-group, this will display the user which transferred the group to PAS. </p>
Enrollment	<p> schoolYear: Link to the academicSession (type: schoolYear) this enrollment belongs to. </p>

Payload	Metadata
	<p>inactiveDate: (student) inactiveDate</p> <p>positions: (teacher) Array of positions for the teacher. Contains the following data:</p> <ul style="list-style-type: none"> positionName: (teacher) positionName positionDescription: (teacher) positionDescription positionPercentage: (teacher) positionPercentage. employmentType: (teacher) employmentType. status: (teacher) employee position status (starting, quit, active). endDate: (teacher) positionEndDate
Course	N/A

6.1 Overview of GroupTypes in Vigilo OAS

The following groups types are available, and can be filtered:

Group	Norwegian name	English name	Norwegian comments
Class	Skole: Klasse	School: Class	Skole: Elevens tilhørighet til klasse/basisgruppe
	Barnehage: Avdeling	Childcare: Department	Barnehage: Barnet tilhørighet til avdeling
	SFO: SFO-modell	Afterschool: AfterSchool model	SFO: Elevens tilhørende SFO-modell i enheten
SubjectGroup	Faggruppe	Subject Group	Faggrupper
PasTestGroup	PAS Prøvegruppe	PAS – test group	Kun for PAS prøvegrupper
LearningSupport	Spesialundervisning	Learning support	Undervisningsgrupper for elever med rett til spesialundervisning hvor det er fattet et enkeltvedtak
SnoGroup	Særskilt norskopplæring	Adapted language education	Undervisningsgrupper for elever med rett til Særskilt norskopplæring hvor det er fattet et enkeltvedtak
GnoGroup	Grunnleggende norskopplæring	Basic Norwegian	Undervisningsgrupper for elever med rett til Grunnleggende Norskopplæring hvor det er fattet et enkeltvedtak
SfoGroup	Skolefritidsordning	After school activity	Gruppe for elever som har SFO-plass, hvor skolene velger å legge «Miljøtimeplaner» med tilhørende leksjoner i Vigilo
SfaGroup	Skolefriavlastning	After school care for special-needs children	Grupper for elever som har SFA plass, hvor skolene velger å legge «Miljøtimeplaner» med tilhørende leksjoner i Vigilo.
TpoGroup	Ekstra/TPO	Extra	Undervisningsgrupper for elever som trenger tilrettelegging/tilpasset opplæring, hvor det ikke er fattet enkeltvedtak
TofGroup	Tospråklig fagopplæring	Bilingual subject teaching	Undervisningsgrupper for elever med rett til tospråklig fagopplæring hvor det er fattet et enkeltvedtak
NativeLanguage	Morsmålsopplæring	Native-language instruction group	Undervisningsgrupper for elever med rett til morsmålsopplæring hvor det er fattet et enkeltvedtak

Group	Norwegian name	English name	Norwegian comments
OtherUsers	Andre brukere	Other users	Benyttes til grupper det en ikke ønsker å ha med elever, f.eks: <ul style="list-style-type: none">• Inspeksjon• Elevsamtaletime• Møtetid• Utleie

Note: For information security reasons two groups types are NOT available through OneRoster API:

- WrittenExamnGroup
- oralExamnGroup

7 Implementation guide

1. Get parents: `/users?filter=role='parent'`
 - a. Get schools: `/schools/`
 - For each school (avoid parallelism):**
 1. Get students `/schools/<orgId>/students`
 2. Get teachers `/schools/<orgId>/teachers`
 3. Get present active school year: `/schools/<orgId>/academicSessions?filter=type='schoolYear' AND status='active'`
 4. Use 'sourceld' from active school year (unique pr. school) to get classes:
`/schools/<orgId>/classes?filter=schoolYearId='<schoolYearId (sourceld) in academicSessions>'`
 5. Use sourceld from active school year to get enrollments:
`/schools/<orgId>/enrollments?filter=schoolYearId='<schoolYearId (sourceld) in academicSessions>'` (you might also include `classType='homeroom'` which will perform better, but you'll only get enrollments for homeroom classes (e.g: "5B"). If `classType='scheduled'` is used, you'll get enrollments for other groups (e.g: "Engelsk 5B").
 6. Use enrollments to filter which teachers and students are active in the different classes.
 7. Find active employees which are not enrolled (e.g admin staff), by finding the teachers (step 2) which have metadata status "active" (**new and better alternative: Check metadata-tag "employeeStatus" for value "active"**) and are not in the list of teachers found in 6.

7.1 How to exclude previous administrative staff in school

To find previous administrative staff in school, find the teachers that are NOT enrolled AND where the metadata-tag employeeStatus = "ended".
For child care the employeeStatus can be used in the same way to identify previous staff in general.
Employees that has from-date in the future will have employeeStatus = "starting".

~~In current HR model in Vigilo all employees will have metadata status = "Active". To sort out previous school employees that should be set inactive in customers IAM systems (Feide and AD catalog), the metadata for POSITIONS has to be included in the filtering process:~~

~~To find inactive administrative staff in school is done like this:~~

~~**Find the teachers that are not enrolled AND has no active/current positions.**~~

~~Note that this method is currently not valid for childcare staff, as the position concept for childcare employees are not a part of current HR model for child care. To handle childcare positions, you'd have to use available metadata tags on employee (e.g. "AD User name") to check which employees are active or not, specially related to employees with part-time employments/positions in different units within the same municipality. Such methods should be used with care and in cooperation between municipality administrator and internal or external IAM partner (Identity and Access Management).~~

7.2 Implementation notes

Subject	Note
startDate and endDate definitions	<p>startDate is always defines as “included”.</p> <p>endDate is from 13.Jan 2022 defined as <u>excluded</u>, meaning that if a school-year has 31.July as it last active day, the endDate of the academicSession will be 1.August.</p> <p>Note also that the data type for startDate end endDate are changed from DateTime to Date, with effect from 19.Dec 2021.</p> <p>The changes are compliant with IMS OneRoster standard.</p> <p>This change affects data-sets where start- and end-dates are relevant, like academic sessions, terms, employee data and position data.</p>
<p>Childcare employee positions info</p> <p>AfterSchool employee positions info</p>	<p>As for 13.Jan 2022 the position info for childcare employees are fictive, meaning that the position “startDate” and “endDate” are the dates from the EMPLOYEE start and end dates, and not from the actual employment/position as registered in HR system.</p> <p>Note that employees in AfterSchool are connected to the corresponding school, and has not separate employment info in AfterSchool context.</p>

8 Endpoints - Childcare extension

Base url: <https://dataporten-api.vigilo.no:4004/api/<tenantId>ims/oneroster/v1p1>

Action	URI	Supported filter *
getAllChildcares	/childcares	
getChildcare	/childcares/<childcareId>	
getUsersForChildcare	/childcares/<childcareId>/users	Identifier status: active tobedeleted
getStudentsForChildcare	/childcares/<childcareId>/students	identifier status: active tobedeleted
getTeachersForChildcare	/childcares/<childcareId>/teachers	identifier
getTeacher	/teachers/<userId>	
getClassesForChildcare	/childcares/<childcareId>/classes/	
getClass	/childcares/<childcareId>/classes/<classId> Alt: /classes/<classId>	
getEnrollmentsForChildcare	/childcares/<childcareId>/enrollments	status: active tobedeleted
getEnrollment	/childcares/<childcareId>/enrollments/<enrollmentId> Alt:/enrollments/<enrollmentId>	

9 Endpoints – AfterSchool extension

Base url: <https://dataporten-api.vigilo.no:4004/api/<tenantId>ims/oneroster/v1p1>

Action	URI	Supported filter *
getAllAfterSchools	/afterschools	
getAfterSchool	/afterschools/<afterSchoolId>	
getUsersForAfterSchool	/afterschools/<afterSchoolId>/users * will only return students	status: active tobedeleted
getStudentsForAfterSchool	/afterschools/<afterSchoolId>/users	status: active tobedeleted
getClassesForAfterSchool	/afterschools/<afterSchoolId>/classes	status: active tobedeleted schoolYearId: sourcedId from academicSessions
getClass	/afterschools/<afterSchoolId>/classes/<classId> Alt: /classes/<classId>	
getEnrollmentsForAfterSchool	/afterschools/<afterSchoolId>/enrollments	status: active tobedeleted schoolYearId: sourcedId from academicSessions
getEnrollment	/afterschools/<afterSchoolId>/enrollments/<enrollmentId> Alt: /enrollments/<enrollmentId>	
getAcademicSessionsForAfterSchool	/afterschools/<afterSchoolId>/academicSessions	status: active tobedeleted
getAcademicSession	/afterschools/<afterSchoolId>/academicSessions/<academicSessionId> Alt: /academicSessions/<academicSessionId>	

10 Endpoints – Application extension

Beta-testing in progress: Changes may come without further notice!

Base url: <https://dataporten-api.vigilo.no:4004/api/<tenantId>ims/oneroster/v1p1>

Note: The payload for afterschool-applications and childcare-applications are structurally different from each other.

Applications endpoints also supports limit & offset parameters, e.g: [/applications?limit=100&offset=200](#)

Action	URI	Supported Query/Filter
getApplicationsForAfterSchools	/afterschools/applications	personId: Query-parameter, ex: ?personId=<sourcedId uid of the person from the oneRoster API> receivedFromTime: Query-parameter, ex: ?receivedFromTime=2021-01-01T08:00:00 receivedToTime: Query-parameter, ex: ?receivedFromTime=2021-02-01T08:00:00
getApplicationsForChildcares	/childcares/applications	personId: Query-parameter, ex: ?personId=<sourcedId uid of the person from the oneRoster API> receivedFromTime: Query-parameter, ex: ?receivedFromTime=2021-01-01T08:00:00 receivedToTime: Query-parameter, ex: ?receivedFromTime=2021-02-01T08:00:00
getApplicationsForAfterSchool	/afterschools/<afterSchoolId>/applications	personId: Query-parameter, ex: ?personId=<sourcedId uid of the person from the oneRoster API> receivedFromTime: Query-parameter, ex: ?receivedFromTime=2021-01-01T08:00:00 receivedToTime: Query-parameter, ex: ?receivedFromTime=2021-02-01T08:00:00
getApplicationsForChildcare	/childcares/<childcareId>/applications	personId: Query-parameter, ex: ?personId=<sourcedId uid of the person from the oneRoster API> receivedFromTime: Query-parameter, ex: ?receivedFromTime=2021-01-01T08:00:00

Action	URI	Supported Query/Filter
		receivedToTime : Query-parameter, ex: ?receivedFromTime=2021-02-01T08:00:00
getApplicationForAfterSchool	/afterschools/applications/<applicationId>	
getApplicationForChildCare	/childcares/applications/<applicationId>	

11 Endpoints – Sessions

Beta-testing in progress: Changes may come without further notice!

Base url: <https://dataporten-api.vigilo.no:4004/api/<tenantId>ims/oneroster/v1p1>

Note: Will return data for the present active schoolyear if no academicSessionId is given.

Action	URI	Supported Query/Filter
getSessionsForStudentInSchool	<code>schools/<schoolId>/students/<studentId>/sessions</code>	week: the weeknumber you want sessions for, like filter=week='18' academicSessionId: sourcedId of the academicSession (schoolyear)
getSessionsForEnrollmentInSchool	<code>schools/<schoolId>/enrollments/<enrollmentId>/sessions</code>	week: the weeknumber you want sessions for, like filter=week='18' academicSessionId: sourcedId of the academicSession (schoolyear)

Since sessions is not currently within scope of IMS OneRoster 1.1 specification, the below table explains the API endpoint in more details:

JSON-tag	Type	Multiplicity	Comment
sessions	core	n/a	
sourcedId	string	1	Id
status	enum	1	'active' for the present schoolyear, 'tobedeleted' for inactive schoolyears (previous or next).

JSON-tag	Type	Multiplicity	Comment
dateLastModified	DateTime	1	
metadata	metadata	0..1	
title	string	1	The title of the subject, corresponding to the courseCode.
year	string	1	The year the session takes place
week	string	1	The week the session takes place Data type might be changed to number without further notice.
courseCode	string	1	The Vigilo-internal code for the subject. Note: This is NOT the KL06/GREP-code for the course for the actual grade. The GREP-code can be found via the class-reference.
startTime	time	1	Start-time of the session
endTime	time	1	End-time of the session
dayOfWeek	number	1	The day of the week the session are held: Mon = 1 .. Fri = 5.
duration	number	1	Duration of the session in minutes.
room	string	0..1	The physical room for the session.
className	string	1	The name of the corresponding class for the session

JSON-tag	Type	Multiplicity	Comment
class (href / sourcedId / type)	Link	1	Link to corresponding class
teacher (href / sourcedId / type)	Link	0..1	Link to user assigned to the session
schoolYear (href / sourcedId / type)	Link	1	Link to the academicSession of type "school-year" for the session

12 Endpoints – Student absence

Beta-testing in progress: Changes may come without further notice!

Base url: <https://dataporten-api.vigilo.no:4004/api/<tenantId>ims/oneroster/v1p1>

Note: One “absence” is the absence for a student during a schoolyear. Use status filter (?filter=status='active') to retrieve for the current schoolyear and not load the service too heavily. Absence for previous and pre-registered (next) schoolyears come with status “tobedeleted” as they are to be considered historical and/or not relevant for the present year. The individual absences of the student (hours/days) may come if you include query-parameter ?includeAbsenceDetails=true, but use this with care as it loads a lot of data and may impact performance. Over-eagerly use of this parameter may cause us to revoke access without notice.

Action	URI	Supported Query/Filter
getStudentAbsencesForStudentIn School	schools/<schoolId>/students/<studentId>/absences	includeAbsenceDetails: true false academicSessionId: sourcedId of the academicSession (schoolyear) status: active tobedeleted
getStudentAbsenceForStudentIn School	schools/<schoolId>/students/<studentId>/absences/<absencId>	includeAbsenceDetails: true false
getStudentAbsencesForSchool	schools/<schoolId>/absences	includeAbsenceDetails: true false academicSessionId: sourcedId of the academicSession (schoolyear) status: active tobedeleted
getStudentAbsencesInSchool	schools/<schoolId>/absences/<absencId>	includeAbsenceDetails: true false
getStudentAbsence	absences/<absencId>	includeAbsenceDetails: true false

Since StudentAbsence is not currently within scope of IMS OneRoster 1.1 specification, the below table explains the API endpoint in more details:

JSON-tag	Type	Multiplicity	Comment
studentAbsences	core	n/a	
sourcedId	String	1	Id of the absence.
status	enum	1	'active' for the present schoolyear, 'tobedeleted' for inactive schoolyears (previous or next).
dateLastModified	DateTime	1	
metadata	metadata	0..1	
totalDaysAbsent	Number	1	Number of days registered, adjusted with numberOfDaysAdjusted
totalMinutesAbsent	Number	1	Number of minutes registered, adjusted with numberOMinutesAdjusted
deductedDaysAbsent	Number	1	Number of absence days that shall be deducted (not to be shown in 10 th grade diploma as absence days)
deductedMinutesAbsent	Number	1	Number of absence minutes that shall be deducted (not to be shown in 10 th grade diploma as absences hours)
adjustedDaysAbsent	Number	1	Manually registered addition or withdrawal of absence days for the given SchoolYear. Affect the totalDaysAbsent number.
adjustedMinutesAbsent	Number	1	Manually registered addition or withdrawal of absence minutes for the given SchoolYear. Affect the totalMinutesAbsent number.

JSON-tag	Type	Multiplicity	Comment
accumulatedDaysAbsent	Number	1	Number of absence days to be shown in 10 th grade diploma as absence days for the given SchoolYear.
accumulatedMinutesAbsent	Number	1	Number of absence minutes to be shown in 10 th grade diploma as absence hours for the given SchoolYear.
user	String	1	Id of the student's unique personId
school	String	1	Id of the schools unique OrgId.
academicSession	String	1	Id of the unique academicSession (school-year),
absenceDetails			
Id	String	1	GUID of the single instance of an absence.
fromTime	DateTime	1	The timestamp for the startTime of the absence. Full-day absence are indicated with time 00:00:00.
toTime	DateTime	1	The timestamp for the endTime of the absence. Full-day absence are indicated with time 00:00:00.
absenceDurationUnit	enum	1	Values: DAYS or MINUTES
absenceDurationValue	Number	1	Note: In Vigilo all student absence are signed "day by day", and absence of Unit = DAYS will hence never be larger the Value = 1.
registeredByUserId	String	1	Id of user who registered the absence.

JSON-tag	Type	Multiplicity	Comment
registeredTime	DateTime	1	
comment	String	0..1	The comment (if any) from subject teacher when absence are registered.
absenceClassification	enum	1	<p>invalid – absence signed as undocumented in Vigilo w/ check-mark “absence”.</p> <p>valid - absence signed as documented in Vigilo w/ check-mark “documentation”.</p> <p>leave - absence signed as leave (permission) in Vigilo.</p> <p>leaveWithDeduction - absence signed as “Leave with deduction” in Vigilo w/ check-mark “deduction”.</p>
signingInfo			Including registered comment by teacher signing the absence.
- comment	String	0..1	The comment (if any) from contact teacher when signing the absence.
- signedTime	DateTime	1	Note that only <u>signed</u> absence are returned in the response for student absenceDetails.
- signedByUserId	String	1	Id of user who signed the absence.
sessionsAffected	Array of strings	0..1	ID's to the sessions corresponding (if any) to the registered period of absence.
classesAffected	Array of numbers	0..1	<p>ID's to the session's corresponding subject classes (if any).</p> <p>Note: If same subject are held in sub-subsequent sessions, the ID's of classes (subjectGroups) might be replicated. This indicated two sessions with the same class. (e.g. double-lesson of mathematics). See below example:</p>

JSON-tag	Type	Multiplicity	Comment
			"classesAffected" : [13002, 13020, 13020, 13038, 13038, 13047]
absenceAdjustments			
comment	String	0..1	Value "imported" if imported absence info. Otherwise comment given when registered absence adjustments.
days	Number	0..1	Can be positive or negative values
minutes	Number	0..1	Can be positive or negative values
registeredByUserId	String	1	ID of user who registered this adjustment
registeredTime	DateTime	1	
Term <Link>	Link	1	Link to the term this adjustment is registered to.