

Vigilo OneRoster implementation

Table of content

1	Change log	2
2	Required	3
3	Restrictions (might change)	3
4	Cache	4
5	Endpoints	5
6	Additional Metadata	8
6.1	Overview of GroupTypes in Vigilo OAS	11
7	Implementation guide	13
7.1	How to exclude previous administrative staff in school	14
8	Endpoints - Childcare extension	15
9	Endpoints – AfterSchool extension	16
10	Endpoints – Application extension	17

1 Change log

- 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.
- 10.11.2121** 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
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	
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 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 gsild: Organization gsild managerFirstName: Organization managerFirstName managerLastName: Organization managerLastName vigold: Organization vigold nationalRegisterId: Organization nationalSchoolRegisterId shortName: Organization pasCode key: Only for tenants: Tenant “key”. ownership: Schools: “Privat” ”Offentlig” ChildCares: “Privat” ”Kommunal”</p>

Payload	Metadata
<p>User</p> <p>Note: 1 person might be several users with the same sourcedId, 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.</p> <p>gender: male female.</p> <p>birthDate: Date of birth (format: yyyy-MM-dd).</p> <p>fromDate: (student) fromDate (Not currently in use!)</p> <p>toDate: (student) toDate (Not currently in use!)</p> <p>Only returned when getting in the context of an organization:</p> <p>inactiveDate: (student) inactiveDate</p> <p>positions: (teacher) Array of positions for the teacher (even for childcare employees). May contain 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 <p>primaryOrg: true false - only on students attending school (not childcare). (User interface: hospiteringselev) Default: true.</p> <p>studentId: N/A: internal use only.</p> <p>clientId: N/A: internal use only.</p> <p>personId: N/A: internal use only (same as sourcedId).</p> <p>schoolTransportFromDate: Fromdate this student is eligible for free school transport</p> <p>schoolTransportEndDate: Enddate this student is eligible for free school transport</p> <p>schoolTransportDescription: School transport description.</p> <p>userIds (not in metadata-tag, but documented here as it's dynamic):</p> <p>adUserIdentifier: ActiveDirectory username.</p> <p>feideUserIdentifier: Feide username.</p> <p>upnUserIdentifier: UPN username.</p>
<p>AcademicSession</p>	<p>status: status of the schoolYear directly from Vigilo. "Nåværende Utgått Pre_registered"</p> <p>org: Link to the Organization of which this academicSession belongs.</p>
<p>Class</p>	<p>schoolYear: Link to the academicSession (type: schoolYear) this class belongs to.</p> <p>pasPeriod: If this instance is a PAS-group, this will display PAS Period.</p>

Payload	Metadata
	<p>pasSubjectCode: If this instance is a PAS-group, this will display PAS SubjectCode.</p> <p>groupType: GroupType from Vigilo. Possible values to date: (Class, LearningSupport, SubjectGroup, SnoGroup, GnoGroup, SfoGroup, SfaGroup, TpoGroup, NativeLanguage, TofGroup, OtherUsers, PasTestGroup)</p> <p>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> <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" and are not in the list of teachers found in 6.

7.1 How to exclude previous administrative staff in school

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).

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=<sourceId 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=<sourceId 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=<sourceId 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=<sourceId 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>	