Vigilo OneRoster implementation

Table of content:

Change log

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.

Required

TenantId
IP of caller (for ip-filter pr Tenant).
Use HTTP-header for authentication: Key: <tenantId>, value: <token given by Vigilo>

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.

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>	

	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> **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>	

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	**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 **gsiId**: Organization gsiId **managerFirstName**: Organization managerFirstName **managerLastName**: Organization managerLastName **vigoId**: Organization vigoId **nationalRegisterId**: Organization nationalSchoolRegisterId **shortName**: Organization pasCode **key**: Only for tenants: Tenant “key”. **ownership**: Schools: “Privat”|”Offentlig” | ChildCares: “Privat”|”Kommunal”

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.	**formOfWrittenNorwegian**: nn | nb. **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!)** Only returned when getting in the context of an organization: **inactiveDate**: (student) inactiveDate **positions:** (teacher) Array of positions for the teacher (even for childcare employees). May contain the following data: **positionName**: (teacher) positionName **positionDescription**: (teacher) positionName **status:** (teacher) employee position status. **endDate**: (teacher) positionEndDate **primaryOrg:** true|false - only on students attending school (not childcare). (User interface: hospiteringselev) Default: true. **studentId:** N/A: internal use only. **clientId:** N/A: internal use only. **personId:** N/A: internal use only (same as sourcedId). **schoolTransportFromDate:** Fromdate this student is eligible for free school transport **schoolTransportEndDate:** Enddate this student is eligible for free school transport **schoolTransportDescription:** School transport description. Deprecated position information has been removed from docs. May still come. Please change to positions structure above. **userIds** (not in metadata-tag, but documented here as it's dynamic): **adUserIdentifier**: ActiveDirectory username. **feideUserIdentifier**: Feide username. **upnUserIdentifier**: UPN username.
AcademicSession	**status**: status of the schoolYear directly from Vigilo. "Nåværende|Utgått|Pre_registered" **org**: Link to the Organization of which this academicSession belongs.

Class	**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.
Enrollment	**schoolYear**: Link to the academicSession (type: schoolYear) this enrollment belongs to. **inactiveDate**: (student) inactiveDate **positions:** (teacher) Array of positions for the teacher. Contains the following data: **positionName**: (teacher) positionName **positionDescription**: (teacher) positionName **status:** (teacher) employee position status. **endDate**: (teacher) positionEndDate DEPRECATED: ~~**positionName**: (teacher) positionName (if several positions; each name divided by character '|')~~ ~~**positionDescription**: (teacher) positionName (if several positions; each name divided by character '|')~~ ~~**endDate**: (teacher) positionEndDate (if several positions; each name divided by character '|')~~ ~~**status:** (teacher) position status. (if several positions; each name divided by character '|')~~
Course	N/A

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 ‘sourceId’ from active school year (unique pr. school) to get classes: /schools/<orgId>/classes?filter=schoolYearId='<schoolYearId (sourceId) in academicSessions>'
   5. Use sourceId from active school year to get enrollments: /schools/<orgId>/enrollments?filter=schoolYearId='<schoolYearId (sourceId) 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 has metadata status “active” and are not in the list of teachers found in 6.