

# Vigilo OneRoster implementation

## Table of content:

<b>Required:</b>	<b>1</b>
<b>Restrictions (might change):</b>	<b>1</b>
<b>Endpoints</b>	<b>2</b>
<b>Additional Metadata</b>	<b>5</b>
<b>Implementation guide</b>	<b>7</b>

---

## Required:

TenantId

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

Use 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.

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

## 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)	<b>type:</b> schoolYear   term <b>status:</b> active   tobedeleted
getAcademicSessionsForSchool	/schools/<schoolId>/academicSessions	<b>type:</b> schoolYear   term <b>status:</b> active   tobedeleted
getAcademicSession	/academicSessions/<academicSessionId> Alt: /schools/<schoolId>/academicSessions/<academicSessionId>	
getAllUsers	/users	<b>Identifier</b> <b>role:</b> parent

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

getClassesForStudent	/school/<schoolId>/student/<studentId>/classes	<b>schoolYearId:</b> <schoolYearId> <b>classType:</b> homeroom   scheduled - OR - <b>groupType:</b> Vigilo-internal value which is presented in metadata tag
getClass	/classes/<classId> Alt: /schools/<schoolId>/classes/<classId>	
getAllCourses	<del>/courses (restricted due to heavy load)</del> In development	
getCoursesForSchool	<del>/schools/&lt;schoolId&gt;/courses/</del> In development	
getCourse	In development	
getAllEnrollments	/enrollments (restricted due to heavy load unless schoolYearId is specified; /enrollments)	<b>schoolYearId:</b> <schoolYearId> <b>status:</b> active   tobedeleted <b>role:</b> teacher   student
getEnrollmentsForSchool	/schools/<schoolId>/enrollments	<b>schoolYearId:</b> <schoolYearId> <b>status:</b> active   tobedeleted <b>role:</b> teacher   student
getEnrollmentsForClassInSchool	/schools/{school_id}/classes/{class_id}/enrollments	<b>status:</b> active   tobedeleted <b>role:</b> 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	<p><b>academicSessions:</b> Link to AcademicSessions for this org.  <b>classes:</b> Link to Classes for this org.  <b>students:</b> Link to Students (user-objects) for this org.  <b>teachers:</b> Link to Teachers (user-objects) for this org.  <b>enrollments:</b> Link to Enrollments for this org.  <b>courses:</b> Link to Courses for this org.  <b>externalOrganizationNumber:</b> OrganizationUnit - externalOrganizationNumber  <b>addressline1:</b> Organization addressline1  <b>addressline2:</b> Organization addressline2  <b>postalCode:</b> Organization postalCode  <b>city:</b> Organization city  <b>latitude:</b> Organization latitude  <b>longitude:</b> Organization longitude  <b>phoneNumber:</b> Organization phoneNumber  <b>email:</b> Organization email  <b>homePageUrl:</b> Organization homePageUrl  <b>gsild:</b> Organization gsild  <b>managerFirstName:</b> Organization managerFirstName  <b>managerLastName:</b> Organization managerLastName  <b>vigold:</b> Organization vigold  <b>nationalRegisterId:</b> Organization nationalSchoolRegisterId  <b>shortName:</b> Organization pasCode  <b>key:</b> Only for tenants: Tenant “key”.</p>

<p>User</p> <p>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><b>formOfWrittenNorwegian:</b> nn   nb.</p> <p><b>gender:</b> male   female.</p> <p><b>birthDate:</b> Date of birth (format: <b>yyyy-MM-dd</b>).</p> <p><b>fromDate:</b> (student) fromDate (<b>Not currently in use!</b>)</p> <p><b>toDate:</b> (student) toDate (<b>Not currently in use!</b>)</p> <p><b>Only returned when getting in the context of an organization:</b></p> <p><b>inactiveDate:</b> (student) inactiveDate</p> <p><b>positions:</b> (teacher) Array of positions for the teacher. Contains the following data:</p> <ul style="list-style-type: none"> <li><b>positionName:</b> (teacher) positionName</li> <li><b>positionDescription:</b> (teacher) positionName</li> <li><b>status:</b> (teacher) employee position status.</li> <li><b>endDate:</b> (teacher) positionEndDate</li> </ul> <p><b>DEPRECATED:</b></p> <p><del><b>positionName:</b> (teacher) positionName (if several positions; each name divided by character ' ')</del></p> <p><del><b>positionDescription:</b> (teacher) positionName (if several positions; each name divided by character ' ')</del></p> <p><del><b>endDate:</b> (teacher) positionEndDate (if several positions; each name divided by character ' ')</del></p> <p><del><b>status:</b> (teacher) employee position status. (if several positions; each name divided by character ' ')</del></p> <p><b>userIds</b> (not in metadata-tag, but documented here as it's dynamic):</p> <p><b>adUserIdentifier:</b> ActiveDirectory username.</p> <p><b>feideUserIdentifier:</b> Feide username.</p> <p><b>upnUserIdentifier:</b> UPN username.</p>
<p>AcademicSession</p>	<p><b>status:</b> status of the schoolYear directly from Vigilo. "Nåværende Utgått Pre_registered"</p> <p><b>org:</b> Link to the Organization of which this academicSession belongs.</p>
<p>Class</p>	<p><b>schoolYear:</b> Link to the academicSession (type: schoolYear) this class belongs to.</p> <p><b>pasPeriod:</b> If this instance is a PAS-group, this will display PAS Period.</p> <p><b>pasSubjectCode:</b> If this instance is a PAS-group, this will display PAS SubjectCode.</p> <p><b>groupType:</b> GroupType from Vigilo. Possible values to date: (Class, LearningSupport, SubjectGroup, SnoGroup, GnoGroup, SfoGroup, SfaGroup, TpoGroup, NativeLanguage, TofGroup, OtherUsers, PasTestGroup)</p> <p><b>pasTransferRequestedByPersonId:</b> If this instance is a PAS-group, this will display the user which transferred the group to PAS.</p>

Enrollment	<p><b>schoolYear:</b> Link to the academicSession (type: schoolYear) this enrollment belongs to.</p> <p><b>inactiveDate:</b> (student) inactiveDate</p> <p><b>positions:</b> (teacher) Array of positions for the teacher. Contains the following data:</p> <ul style="list-style-type: none"> <li><b>positionName:</b> (teacher) positionName</li> <li><b>positionDescription:</b> (teacher) positionName</li> <li><b>status:</b> (teacher) employee position status.</li> <li><b>endDate:</b> (teacher) positionEndDate</li> </ul> <p><b>DEPRECATED:</b></p> <p><del><b>positionName:</b> (teacher) positionName (if several positions; each name divided by character ' ')</del></p> <p><del><b>positionDescription:</b> (teacher) positionName (if several positions; each name divided by character ' ')</del></p> <p><del><b>endDate:</b> (teacher) positionEndDate (if several positions; each name divided by character ' ')</del></p> <p><del><b>status:</b> (teacher) position status. (if several positions; each name divided by character ' ')</del></p>
Course	N/A - Under development.

## Implementation guide

1. Get parents: /users?filter=role='parent'
  - a. Get schools: /schools/
    - For each school:**
      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>'
      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.