Guidelines for OneRoster API Consumers

Last Modified on 12/01/2025 10:33 am CST

This article is intended for use by members of the Digital Learning Partner Program and is not designed for general Campus users.

The Campus OneRoster API is used to integrate core elements from the Campus SIS with other 3rd party systems. The following recommendations support successful implementations and mitigate performance concerns.

General Synchronization Recommendations

- Perform full syncs weekly.
- Perform differential syncs on a nightly basis, filtering on dateLastModified values.
- Perform all syncs during off-peak hours, as defined by the district.
    - Consumers should ensure that syncs complete prior to the end of the off-peak window.
- Utilize Campus-defined core endpoints and override the default page size to optimize performance (See the section below).
- Make no concurrent requests during peak hours (serial requests only).
    - Make no more than 5 concurrent requests during off-peak hours.
    - Districts should coordinate with their vendors to ensure that they are not all hitting the same objects at the same time.

Optimizing Data Pulls

The following is a recommended sequence of endpoint calls and pagination limits to pull the data in the most efficient way. If you are encountering performance issues, please be sure that district support contacts are aware of the full request URL with district app and API path, including parameters values for sorting, filtering, paging...etc.

Please note that Campus has exposed additional endpoints that are not included in the original OneRoster v1.0 specifications. These endpoints are intended to supplement or replace contextual or aggregate endpoints for the purpose of syncing large sets of data. For example, Results only exist in the specification with respect to a class (e.g. /learningdata/v1/classes/<classSourcedId>/results). Campus also breaks down User and Enrollment types into separate endpoints. By using these endpoints, a sync's performance can be improved by loading more records in a single request and ultimately issuing less requests.

See the OneRoster 1.1 Data Models and OneRoster API Documentation (OAuth 2) articles for more information.

1. Orgs (Districts and Schools)
   - orgs [limit=50K]
2. AcademicSessions
   - academicSessions [limit=50k]
   - terms [limit=50K]
   - gradingPeriods [limit=50K]
3. Demographics (Optional)
   - /learningdata/v1/demographics [limit=50K]
4. Users
   - users [limit=50K]
   - students [limit=50K]
   - teachers [limit=50K]
5. Courses and Classes
   - courses [limit=50K]
   - classes [limit=50K]
6. Enrollments
   - enrollments [limit=50K])
   - schools/{schoolSourcedId}/classes/{classSourcedId}/enrollments - returns only active enrollments for class
7. Categories
   - /learningdata/v1/categories [limit=100]
8. LineItems (Utilize core endpoint where possible, else /learningdata/v1/classes/<classSourcedId>/lineitems (for each class) [limit=100])
   - /learningdata/v1/lineitems [limit=50K]
9. Results (Utilize core endpoint where possible, else /learningdata/v1/classes/<classSourcedId>/results (for each class) [limit=100])
   - /learningdata/v1/results [limit=50K]
10. Score Scales (1.2 only)
    - scoreScales