ED-FI RFC 13 - ENROLLMENT API

Ed-Fi Request for Comment: 13 ENROLLMENT API
Product: Ed-Fi Data Standard v3.0
Affects: Ed-Fi API Design & Implementation Guidelines (and Ed-Fi ODS / API)
Obsoletes: --
Obsoleted By: RFC 19 - ENROLLMENT API FOR SUITE 3
Status: Inactive

Ed-Fi Alliance
July 5, 2018

Synopsis

The RFC describes a set of read-only REST APIs for transfer of basic student and teacher demographic data and enrollment data for K–12 education. The API is intended to allow source systems to provide the basic data needed to provision a broad range of online teaching and learning applications, and to do so using resource models that align with the Ed-Fi Unifying Data Model 3.0.

The APIs provide enrollment data for sections, schools and local education agencies. The API is limited to basic user and roster data and anticipates a model where other APIs would provide related information such as bell schedules, course or curriculum data for sections, and so forth.

The API resources include only HTTP/HTTPS bindings and the data returned is described only in JSON format. The API is intended to provide options useful in an Web-based architecture that requires a read-only "pull" model of data exchange: a model in which a client target system uses HTTP GETs as the mechanism to implement a unidirectional data exchange between a client system and a data source system that hosts the API surface.

Discussion

Many teaching and learning apps require a basic set of demographic and enrollment data to function. The responsibility for populating such data often falls to individual teachers. In order to allow many teaching and learning apps to be integrated into the classrooms, teachers are required to enter basic student data, arrange students into rosters, and create student accounts. As students enter and leave sections or schools, the responsibility for updating this data also falls to teachers. Unfortunately, as many teachers have little time to tend to such tasks, many online teaching and learning tools remain underused. Correspondingly, when IT departments assume this responsibility, they can quickly find that providing manual, ad-hoc integrations is costly and does not scale as an integration strategy.

Resource Models and Architecture Use Cases

The Enrollment API is designed to allow flexible options for systems to exchange the data necessary to take on typical enrollment data entry tasks. The goal is to reduce or eliminate the burden on teachers to provide manual data integration of this basic information. The API surface (i.e., the resources exposed) provide options for directly accessing a set of data entities, including:

- Local education agencies
- Schools
- Sections
- Students
- Staff

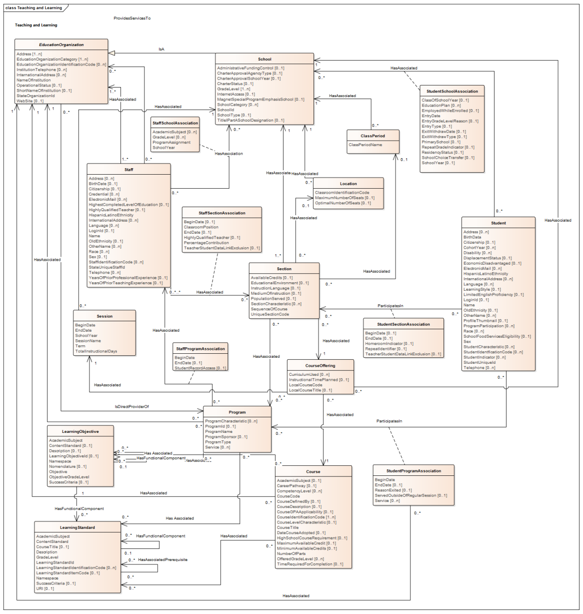
These resources are designed to support a variety of integration scenarios, including:

- **Daily Synchronization.** This is a model in which a client app performs a daily synchronization of data for one or more schools or districts. In this case, a large daily sync of section data is needed. The section resource provides a collection of IDs that reflect the enrollment in that section for students and staff.
- **Granular, Trigger-Based Synchronization.** This is a model in which synchronization is performed at a granular level, for example through an event-based trigger (e.g., a section is synchronized if and when the first student in that section logs in each day). In this case, the `/section` resource can be queried using the resource ID and the results used to check for new enrollments.

Relation to Ed-Fi Unifying Data Model

All resource and field names are derived from and consistent with the Ed-Fi Unifying Data Model and the Ed-Fi Data Handbook. Consistent with the Ed-Fi REST API Guidelines, the resources have an alternate, resource key `"id"`, allowing for ids to be referenced using common REST practice.

The Ed-Fi Teaching and Learning domain is pictured below to help new users understand the relationships.



The Ed-Fi UDM 2.0 Teaching and Learning Domain *(click to expand)*

Resource Paths

Resource paths available in the API are as follows.

- Each path provides the resource or a collection of resources of the types that complete the path. e.g., `/schools/{id}/students` provides a collection of `student` resources.
- If an `id` is provided in the path, that `id` is used to identify the single resource with the ID of the resource type it follows. e.g., for `/schools/{id}/students,` the `school` referred to is the one that has the ID corresponding to `id`.
- The collection provided is one that aggregates the resources associate with the immediately proceeding entity. e.g., `/schools/{id}/students` provides a collection of `student` resources associated with the school identified by `id`.

```
/localEducationAgencies
/localEducationAgencies/{id}
/localEducationAgencies/{id}/schools
/localEducationAgencies/{id}/staffs
/localEducationAgencies/{id}/students
/localEducationAgencies/{id}/sections
/localEducationAgencies/{id}/sectionEnrollment

/schools
/schools/{id}
/schools/{id}/staffs
/schools/{id}/students
/schools/{id}/sections
/schools/{id}/sectionEnrollment

/students
/students/{id}
/students/{id}/schools
/students/{id}/teacher
/students/{id}/sections
/students/{id}/sectionEnrollment

/staffs
/staffs/{id}
/staffs/{id}/schools
/staffs/{id}/students
/staffs/{id}/sections
/staffs/{id}/sectionEnrollment

/sections
/sections/{id}
/sections/{id}/schools
/sections/{id}/staffs
/sections/{id}/students
/sections/{id}/sectionEnrollment
```

Requirements

Implementations of the Assessment Data Collection API must follow the Ed-Fi API Design & Implementation Guidelines for Ed-Fi Technical Suite 3. The key terms under the "Requirement Levels" section from that document carry over to this document.

API Specification

Implementations *must* expose all resources and operations (i.e., HTTP verbs), with all the configurations of required parameters and return values, as indicated in the Open API specification attached to this document.

A visual display of that specification is also attached. Please note that the visual specification is only for ease of reference and the Open API specification (in JSON) is authoritative if there are differences.

Note: HTTP response codes usage follows requirements in the Ed-Fi API Design & Implementation Guidelines under the section "Response Codes." Because Open API files requires at least one response code to validate, the specification attached usually contains only one response for each API path. To eliminate duplication, the response codes were removed from the Open API spec even though they are required for conformance.

Enumerations (Descriptors)

This standard defines a cross-sector context for enrollment data, and so implementations must use only the enumeration sets published as part of the Ed-Fi Data Standards v3, and available in the Alliance Data Standard GitHub repo Descriptors branch and folder.

Field Selection

Field selection is *required* in the API specification to minimize the overall size of the returned entities.

Selection works by appending the fields required in the URI querystring using the fields parameter.

- Values for the fields parameter are comma-separated
- If the `fields` parameter is missing, all data is returned. If included, only included fields are returned.
- References must use a '.' to create a path to elements deeper in the model hierarchy
- If a value in the fields parameter references a JSON object or collection, all fields are returned by default. To return particular fields, fields must be listed in parentheses '()'. For clarity, an asterisk '*' can be used within parentheses as a wildcard to return all fields for that resource or collection of resources.

Field Selection Examples

Field Selection Example 1. Returns only the elements `id, firstName, lastName and studentUniqueId` for the `student` resource

```
/enrollment/students?fields=id,firstName,lastName, studentUniqueId
```

Field Selection Example 2. Returns a collection of `section` resources with two fields (`id` and `sectionUniqueCode`) and the embedded resource `sessions` with only the `schoolYear` field provided

```
 /enrollment/sections?fields=id,sectionUniqueCode,session(schoolYear)
```

Pagination

Pagination is implemented by appending two parameters to the querystring: `limit` and `offset`.

- `limit` specifies how many resources to return
- `offset` specifies the starting position for the collection of resource results to return, often using in supporting pagination for clients

The API does not require ordering, but does require that identical queries return resources in the same order, unless there have been changes to the underlying data between subsequent calls.

Pagination Example

Pagination Example 1. Returns 25 items located at positions 51-75 in the collection

```
 /enrollment/students?limit=25&offset=50
```

Additional Information and Requirements

- Implementations *must* ensure that the Resource IDs provided are unique across resources of that type within the implementing (source) system. It is *recommended* that the Resource IDs returned be a GUID or other guaranteed unique value so as to reduce possible issues with lookups and to improve security.
- All item definitions *must* follow the definitions according to the Ed-Fi UDM Element listed. Those definitions can be found in the Unifying Data Model - v3.0 Handbook
    - If the element has a parent element, the cardinality of the child element is to be understood as applicable if and only if the parent element is included in the resource model.
- Dates must conform to a YYYY:MM:DD format, consistent with ISO 8601 format: http://tools.ietf.org/html/rfc3339.