ED-FI RFC 2 - ASSESSMENT DATA COLLECTION API (Inactive)

Ed-Fi Request for Comment: 2
Product: Ed-Fi Data Standard v2.0
Affects: Ed-Fi Data Standard - Ed-Fi API Design & Implementation Guidelines
Obsoletes: --
Obsoleted By: Ed-Fi RFC 8 - Assessment Outcomes Management API
Status: Inactive

Ed-Fi Alliance
June 7, 2016

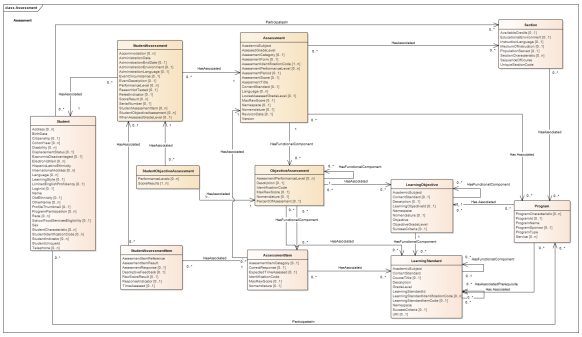
Synopsis

The RFC describes a REST API for exchange of assessment metadata and assessment results. The API is intended to provide structures principally useful for Web-based architectures in which a client source system uses HTTP POST or PUT as the mechanism to allow a source system to push data to and manage data on a target system implementing the API endpoints/resources. This approach is designed to provide a granular "data collection" interface designed to accept data transactionally. The resource models for the data (in JSON) are designed to be compatible with the Ed-Fi Unifying Data Model 2.0.

This RFC is complemented by ED-FI RFC 3 - ASSESSMENT API (Inactive), which provides REST API resources intended for use in a "pull" model of data exchange relying on HTTP GETs.

Discussion

The REST API design directly follows the model implemented as part of the Ed-Fi Operational Data Store and API 2.0 (i.e., the Ed-Fi OPS / API). The data model for API resources are consistent with the Ed-Fi Unifying Data Model v2.0 (UDM), and the API includes many optional resources and resource references that are also consistent with the UDM. The Assessment Domain in the Ed-Fi UDM v2.0 is principally concerned with portability and interoperability of student assessment results, and not with portability of assessment instruments themselves, for which other standards like Question and Test Interoperability® (QTI®) exist. Accordingly, the assessment metadata provided is designed principally to convey information critical in the interpretation of student assessment results, and is not designed to allow an assessment instrument to be "played" on multiple systems.



The Ed-Fi UDM 2.0 Assessment Domain *(click to expand)*

The Assessment Data Collection API consists of a set of API resources including:

- `assessment`
- `objectiveAssessment`
- `assessmentItem`
- `studentAssessment`

The first three resources describe the assessment instrument (i.e., the test, quiz, or other assessment type that was scored) and the last contains data on individual students' performance on a given assessment instrument.

API Uses

The REST resources defined by this API are intended primarily for data collection or synchronization where transactional integration is desirable; that is, uses where individual transactions are delivered in real-time or on a frequent basis from a source system to a target system, as data is added to or modified in the source system (as opposed to a batch updated model). As such, the data flow in this model is intended to be unidirectional, with the source system acting as a system of record, posting data to the target platform where the API resources are implemented. Consistent with this use, the HTTP operations available only create, update, or delete a single resource at a time. This API specification defines no bulk operations or operations that act on collections of resources.

The intent is that the data models used in this API's resources will continue to align with the Ed-Fi UDM through subsequent versions in order to remain consistent with other APIs conforming to the Ed-Fi UDM.

While the Ed-Fi assessment resource models have many components and associations, many interactions will be quite simple. The assessment UDM model exists to support a wide range of use cases, and therefore many elements are optional in simple use cases. See the Examples section below.

Resources and Methods

Each resource can be referenced by two keys, a natural key and a Resource ID. The Resource ID appears as the top level id field in the resource model, and is a string. The Resource ID provides for compatibility with standard REST design where a simple resource identifier (URI) is the core mechanism for identifying individual resources. The natural key serves to define uniqueness through business values, often through a key composed of multiple fields. The uniqueness provided by the natural key assists in several ways, including supporting data quality through key unification, as key fields are shared across entities, and through providing lookup values.

The natural keys for each resource are as follows:

Resource	Natural Keys
`assessment`	• `title` • `assessedGradeLevelDescriptor` • `academicSubjectDescriptor` • `version`
`studentAssessment`	• `title` • `assessedGradeLevelDescriptor` • `academicSubjectDescriptor` • `version` • `studentUniqueId` • `administrationDate`
`objectiveAssessment`	• `title` • `assessedGradeLevelDescriptor` • `academicSubjectDescriptor` • `version` • `identificationCode`
`assessmentItem`	• `title` • `assessedGradeLevelDescriptor` • `academicSubjectDescriptor` • `version` • `identificationCode`

As is evident in this structure, the natural keys for entities associated with an assessment include the key for the assessment resource. This pattern is common in data implementations based on the Ed-Fi UDM.

Note that various data exchange contexts may require use of one or the other — or both — key structures, depending on the particular needs of the data exchange context and the client systems involved. In any case, the natural key fields are required for resource creation. Those fields also cannot be modified if a resource is referenced by Resource ID; the object must be deleted and re-added to modify the natural key if the client references it using the Resource ID.

The API consists of a set of HTTP verbs for each resource, GET, POST, PUT, and DELETE, which are used to implement CRUD operations on the resource. Because there are two key systems for referencing entities, POST and PUT operations give priority to one or the other so as to avoid conflicting ways to resolve each reference.

Examples

Example 1. Posting a new assessment

```
POST /assessments
{
  "title": "Second Week Reading Benchmark",
  "assessedGradeLevelDescriptor": "Third grade",
  "academicSubjectDescriptor": "English Language Arts",
  "version": "1",
  "maxRawScore": "100"
}
```

Example 2. Get a student assessment using the resource natural key

```
GET /studentAssessments?studentUniqueId=605531&assessmentTitle=ACT
&academicSubjectDescriptor=English&assessedGradeLevelDescriptor=Twelfth%
20grade
&version=1&administrationDate=2010-12-01
```

Example 3. Update a student assessment using the resource surrogate key

```
PUT /studentAssessments/1d3eeead9223442a98f7061e8c5f5330
{
  "assessmentReference": {
    "title": "ACT",
    "assessedGradeLevelDescriptor": "Twelfth grade",
    "academicSubjectDescriptor": "English",
    "version": 1,

  },
  "studentReference": {
    "studentUniqueId": "605531"
  },
  "administrationDate": "2010-12-01",
  "scoreResults": [
    {
      "assessmentReportingMethodType": "ACT score",
      "result": "29",
      "resultDatatypeType": "Integer"
    }
  ],
}
```

Specification Content

Implementations of the Assessment Data Collection API must conform to the REST API Design & Implementation Guidelines v2.1. The key terms under the "Requirement Levels" section from that document carry over to this document.

API Specification

Implementations *must* expose the all resources and operations (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 REST API Design & Implementation Guidelines v2.1 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

Additional Requirements

- Implementations *must* ensure that the Resource IDs assigned via a POST are unique across resources of that type within the implementing system; that is the target system must provide a unique value each time a new resource is created. It is *recommended* that the Resource IDs returned from the POST operation 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 Data Handbook for the Core XSD v2.0
    - 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.

Examples - Resources

Examples

Example 1: Populated assessment

Resource: assessment (populated)


```
{
  "id": "5e408a28f7b64a64ba9797c312a59924",
  "title": "4th Grade Reading 2nd Six Weeks 2012-2013 Spanish",
  "assessedGradeLevelDescriptor": "Fourth grade",
  "academicSubjectDescriptor": "English Language Arts",
  "version": 5382,
  "categoryDescriptor": "Benchmark test",
  "revisionDate": "2012-10-30T00:00:00",
  "maxRawScore": 10,
  "namespace": "http://edfi.org",
  "contentStandard": {
    "title": "State Essential Knowledge and Skills",
    "authors": []
  },
  "identificationCodes": [
    {
      "assessmentIdentificationSystemDescriptor": "Test Contractor",
      "identificationCode": "7d659463-4323-4ef7-82f0-98b72246dd31"
    }
  ],
  "languages": [],
  "performanceLevels": [],
  "programs": [],
  "scores": [],
  "sections": []
}
```


Example 2: Populated studentAssessment

Resource: studentAssessment (populated)


```
{
  "id": "dda099447f2944dda7e94890091b90e9",
  "assessmentReference": {
    "title": "ACT",
    "assessedGradeLevelDescriptor": "Twelfth grade",
    "academicSubjectDescriptor": "English",
    "version": 1,
    "link": {
      "rel": "Assessment",
      "href": "/assessments?
title=ACT&assessedGradeLevelDescriptor=Twelfth+grade&academicSubjectDescr
iptor=English&version=1"
    }
  },
  "studentReference": {
    "studentUniqueId": "605387",
    "link": {
      "rel": "Student",
      "href": "/students?studentUniqueId=605387"
    }
  },
  "administrationDate": "2010-12-01T00:00:00",
  "serialNumber": "0",
  "administrationLanguageDescriptor": "English",
  "administrationEnvironmentType": "Testing Center",
  "whenAssessedGradeLevelDescriptor": "Eleventh grade",
  "accommodations": [],
  "items": [],
  "performanceLevels": [],
  "scoreResults": [
    {
      "assessmentReportingMethodType": "Scale score",
      "result": "24",
      "resultDatatypeType": "Integer"
    }
  ],
  "studentObjectiveAssessments": []
}
```


key structures