Webinar: What’s New in Ed-Fi v4.0 and API 6.0

September 21, 2022
Welcome!

• 30-minute webinar, 10-min Q&A
• Ask questions any time through the chat or Q&A feature on the Webex interface.

What We'll Cover

• Overview of Data Standard Changes in v4.0a
• Summary of changes in ODS/API v6.0
Caroline Kazmierski
Director of Marketing Communications

Sayee Srinivasan
Solutions Architect, SEA

Vinaya Mayya
Software Development Lead
Data Standard Changes in v4.0a
Summary of Changes

• Data Standard 4 is targeted at deployment starting in the 2023-24 school year. Data Standard v4 (current version 4.0-a) introduces important updates to 3 domains of the Ed-Fi data model:
  • Student Academic Report (Grade & Gradebook area only)
  • Assessment
  • Finance
• Data Standard 4 introduces the new concept of managing descriptor mappings as data entities
New Entity Descriptor Mapping

{
    "value": "AlgebraII",
    "namespace": "uri://grandbendisd.edu/AcademicSubjectDescriptor",
    "mappedValue": "ADVMATH",
    "mappedNamespace": "uri://somestate.edu/AcademicSubjectDescriptor",
    "modelEntities": [
        {
            "modelEntityDescriptor": "uri://ed-fi.org/ModelEntity#Staff"
        }
    ]
}
“Current Grade”

DATASTD-1696

Current Grade Capture

• Common for systems to capture the current grade and exchange the grade for a grading period that is NOT yet completed.

• Write the “current grade” to the Grade entity for the appropriate GradingPeriod with a GradeType = “Grading Period” with a CurrentGradeIndicator flag = True with a CurrentGradeAsOfDate.
## Grade Entity Changes to capture Current Grade with an asofdate

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
<th>F</th>
<th>G</th>
<th>H</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>GradeType</strong></td>
<td>Reference</td>
<td>object</td>
<td>MetaEd DSL Datatype: DescriptorProperty</td>
<td>Yes</td>
<td>required</td>
<td></td>
<td>The type of grade reported (e.g., Exam, Final, Grading Period).</td>
</tr>
<tr>
<td><strong>GradingPeriodGradingPeriod</strong></td>
<td>Reference</td>
<td>object</td>
<td>MetaEd DSL Datatype: DomainEntityProperty</td>
<td>Yes</td>
<td>required</td>
<td></td>
<td>Identifies the grading period for the Grade.</td>
</tr>
<tr>
<td><strong>StudentSectionAssociation</strong></td>
<td>Reference</td>
<td>object</td>
<td>MetaEd DSL Datatype: AssociationProperty</td>
<td>Yes</td>
<td>required</td>
<td></td>
<td>Relates the student and Section associated with the Grade.</td>
</tr>
<tr>
<td><strong>DiagnosticStatement</strong></td>
<td>String</td>
<td>string</td>
<td>MetaEd DSL Datatype: SharedStringProperty</td>
<td>SQL Recommended Datatype: VARCHAR(124)</td>
<td>No</td>
<td>optional</td>
<td>A statement provided by the teacher that provides information in addition to the grade or assessment score.</td>
</tr>
<tr>
<td><strong>LearningStandardGrade</strong></td>
<td>Reference</td>
<td>object</td>
<td>MetaEd DSL Datatype: CommonProperty</td>
<td>No</td>
<td>optional</td>
<td>collection</td>
<td>A collection of learning standards associated with the grade.</td>
</tr>
<tr>
<td><strong>LetterGradeEarned</strong></td>
<td>String</td>
<td>string</td>
<td>MetaEd DSL Datatype: SharedStringProperty</td>
<td>SQL Recommended Datatype: VARCHAR(20)</td>
<td>No</td>
<td>optional</td>
<td>A final or interim (grading period) indicator of student performance in a class as submitted by the instructor.</td>
</tr>
<tr>
<td><strong>NumericGradeEarned</strong></td>
<td>Number</td>
<td>number</td>
<td>MetaEd DSL Datatype: SharedDecimalProperty</td>
<td>SQL Recommended Datatype: DECIMAL(9, 2)</td>
<td>No</td>
<td>optional</td>
<td>A final or interim (grading period) indicator of student performance in a class as submitted by the instructor.</td>
</tr>
<tr>
<td><strong>PerformanceBaseConversion</strong></td>
<td>Reference</td>
<td>object</td>
<td>MetaEd DSL Datatype: DescriptorProperty</td>
<td>No</td>
<td>optional</td>
<td></td>
<td>A conversion of the level to a standard set of performance levels.</td>
</tr>
<tr>
<td><strong>CurrentGradeIndicator</strong></td>
<td>Boolean</td>
<td>boolean</td>
<td>MetaEd DSL Datatype: BooleanProperty</td>
<td>SQL Recommended Datatype: BOOLEAN</td>
<td>No</td>
<td>optional</td>
<td>An indicator that the posted grade is an interim &quot;current&quot; grade for the grading period and not the final grade for the CurrentGradeAsOfDate.</td>
</tr>
<tr>
<td><strong>CurrentGradeAsOfDate</strong></td>
<td>Date</td>
<td>number</td>
<td>MetaEd DSL Datatype: DateProperty</td>
<td>SQL Recommended Datatype: DATE</td>
<td>No</td>
<td>optional</td>
<td>As-Of date for a grade posted as the current grade.</td>
</tr>
</tbody>
</table>
## AssessmentPeriod in StudentAssessment

<table>
<thead>
<tr>
<th>On /assessments</th>
<th>On /studentAssessments</th>
</tr>
</thead>
<tbody>
<tr>
<td>&quot;periods&quot;: [</td>
<td>&quot;period&quot;: {</td>
</tr>
<tr>
<td>{</td>
<td>&quot;assessmentPeriodDescriptor&quot;: &quot;string&quot;,</td>
</tr>
<tr>
<td>&quot;assessmentPeriodDescriptor&quot;: &quot;string&quot;,</td>
<td>&quot;beginDate&quot;: &quot;2022-08-16&quot;,</td>
</tr>
<tr>
<td>&quot;beginDate&quot;: &quot;2022-08-16&quot;,</td>
<td>&quot;endDate&quot;: &quot;2022-08-16&quot;</td>
</tr>
<tr>
<td>&quot;endDate&quot;: &quot;2022-08-16&quot;</td>
<td>}</td>
</tr>
<tr>
<td>]</td>
<td>}</td>
</tr>
</tbody>
</table>
Time spent in an Assessment

AssessedMinutes in StudentAssessment and in StudentObjectiveAssessment

<table>
<thead>
<tr>
<th>AssessedMinutes</th>
<th>Number</th>
<th>number</th>
<th>MetaEd DSL Datatype: IntegerProperty</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>SQL Recommended Datatype: INT</td>
</tr>
<tr>
<td></td>
<td>No</td>
<td>optional</td>
<td>Reported time student was assessed in minutes.</td>
</tr>
</tbody>
</table>
Other Assessment Domain Changes

- Removal of deprecated elements
  - LearningObjective and all references to that.
  - PerformanceLevelMet Boolean from PerformanceLevel
- ScoreResult changed to an optional collection on StudentObjectiveAssessment
- Remove ambiguities in mapping of performance levels in complex assessments.
  - `AssessmentPerformanceLevel.PerformanceLevelIndicatorName` (as optional)
  - `PerformanceLevel.PerformanceLevelIndicatorName` (as optional)
How to capture not tested student’s information in StudentAssessment

- StudentAssessment.AdministrationDate is not part of the key – Move administration date to optional and update doc to say will be null if not there
Finance Domain is in Core Standard
GradebookEntry Key Structure Change

• The current key structure of DateAssigned, GradebookEntryTitle, and Section (reference) is volatile and not reliable

• The work on the LMS Toolkit found that that these elements of the StudentGradebookEntry key are not reliably available in LMS systems
GradebookEntry

• Current key: DateAssigned, GradebookEntryTitle, Section (reference)

• New GradebookEntry key (DATASTD-1599)
  • Alternative #1 is to use Partial Surrogate Key pattern: GradebookEntryIdentifier + Namespace; Section (reference) not part of key but mandatory
    • Assumes SIS or LMS has a unique ID for each Gradebook Entry across all sections and all school years

• Follows identifier pattern for assessments

• This is a breaking change – but… gradebook domain has not been widely implemented
ODS / API v6.0

- Enhancements to Change Queries
- Delete Tracking
- Combining Authorization Strategies
- Ownership Based Authorization
- Support For External Cache
- Technology Upgrade
Enhancements to Change Queries
Delete Tracking
Change Queries Feature

• Change queries feature allows client systems to narrow requests for data to only data that has changed since a specified point in time without having to pull complete data set.
• First released with ODS/API v3.1 (Dec 2018).
• Initial use cases:
  • Analytics vendors, data warehouse product vendors to stay in sync with source ODS
  • SIS vendors in their reconciliation process
Change Queries API

API clients could use a query syntax to request newer changes on any resource
GET /data/v3/ed-fi/studentProgramAssociations?minChangeVersion=234378&offset=0&limit=25

API clients could get deletes on any resource

```json
[
  {
    "id": "cfa3fe8edf44c50b95e1808e616460a",
    "changeVersion": 234550
  }
]
```
Enhancements to Change Queries Delete Tracking

- Includes natural keys in tracked deletes results
- Adds authorization-based filtering of tracked deletes
- Fixes tracking deletes of derived resources
- Adds support for tracking natural key changes
/deletes


[
  {
    "id": "cfa3fea8edf44c50b95e1808e616460a",
    "changeVersion": 234550,
    "keyValues": {
      "beginDate": "2021-08-30",
      "educationOrganizationId": 255901,
      "programEducationOrganizationId": 255901,
      "programName": "Bilingual",
      "programTypeDescriptor": "uri://ed-fi.org/ProgramTypeDescriptor#Bilingual",
      "studentUniqueId": "604854"
    }
  }
]
/keychanges

GET /data/v3/ed-fi/classPeriods/keyChanges?minChangeVersion=104030&offset=100&limit=500

```json
[
  {
    "id": "e0f8848b54ad4bdddbe81e96737ed7b46",
    "changeVersion": 104036,
    "oldKeyValues": {
      "classPeriodName": "ORIGINAL",
      "schoolId": 255901001
    },
    "newKeyValues": {
      "classPeriodName": "UPDATE 2",
      "schoolId": 255901001
    }
  }
]```
Schema changes

- Tracked via tombstone tables
- Via a delete trigger on the root table, the old and new primary key, resource id and the change version are stored in the tombstone tables
Combining Authorization Strategies
Combining Authorization Strategies

• In previous versions: Resource-Action is authorized with one authorization strategy:
  • NoFurtherAuthorizationRequires
  • NamespaceBased
  • Relationship-based strategies:
    • Relationships with Education Organizations and People
    • Relationships with Education Organizations only
    • Relationships with People only
    • Relationships with Students only
    • Relationships with Students only (through StudentEducationOrganizationResponsibilityAssociation)
    • Relationships with Education Organizations and People (including deletes

Create and Manage API Security Metadata Declaratively
Declarative view

<SecurityMetadata>
  <Claims>
    <Claim name="http://ed-fi.org/ods/identity/claims/program-eligibility/StudentSpecialEducationProgramEligibilityAssociation">
      <DefaultAuthorization>
        <Action name="Create" authorizationStrategy="RelationshipsWithEdOrgsAndPeople"/>
        <Action name="Read" authorizationStrategy="RelationshipsWithEdOrgsAndPeople"/>
        <Action name="Update" authorizationStrategy="RelationshipsWithEdOrgsAndPeople"/>
        <Action name="Delete" authorizationStrategy="RelationshipsWithEdOrgsAndPeople"/>
      </DefaultAuthorization>
    </Claim>
  </Claims>
</SecurityMetadata>
New Use Cases

SPED Eligibility and several others listed in ODS-4846 need more than one authorization strategy in place. Previous versions required database and code changes.
EdFi_Security
<SecurityMetadata>
  <Claims>
    <Claim name="http://ed-fi.org/ods/identity/claims/program-eligibility/StudentSpecialEducationProgramEligibilityAssociation">
      <DefaultAuthorization>
        <Action name="Create" />
        <AuthorizationStrategies>
          <AuthorizationStrategy name="NoFurtherAuthorizationRequired" />
        </AuthorizationStrategies>
        <Action name="Read" />
        <AuthorizationStrategies>
          <AuthorizationStrategy name="RelationshipsWithEdOrgsAndPeople" />
          <AuthorizationStrategy name="RelationshipsWithStudentsOnlyUsingEnrollmentsAndResponsibility" />
        </AuthorizationStrategies>
        <Action name="Update" />
        <AuthorizationStrategies>
          <AuthorizationStrategy name="RelationshipsWithEdOrgsAndPeople" />
          <AuthorizationStrategy name="RelationshipsWithStudentsOnlyUsingEnrollmentsAndResponsibility" />
        </AuthorizationStrategies>
        <Action name="Delete" />
        <AuthorizationStrategies>
          <AuthorizationStrategy name="RelationshipsWithEdOrgsAndPeople" />
          <AuthorizationStrategy name="RelationshipsWithStudentsOnlyUsingEnrollmentsAndResponsibility" />
        </AuthorizationStrategies>
      </DefaultAuthorization>
    </Claim>
  </Claims>
</SecurityMetadata>
Ownership Based Authorization
### Use Case

<table>
<thead>
<tr>
<th>State</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mesa USD</td>
</tr>
<tr>
<td>Chandler USD</td>
</tr>
</tbody>
</table>

**GET** | **POST** | **PUT** | **DELETE** | **Comment**
---|---|---|---|---|
Chandler USD | Y | Y | Y | Y | Chandler USD has CRUD access to all schools under it
Mesa USD | Y | Y | Y | Y | Mesa USD has CRUD access to all schools under it
Deveraux | Y | Y | Y | Y | Both Chandler and Mesa USD have CRUD access to Deveraux data to be able to submit their students
Ownership Based Authorization

- Ownership-based authorization is a **configurable optional feature**:

```json
"ApiSettings": {
  ...
  "Features": [
    {
      "Name": "OwnershipBasedAuthorization",
      "IsEnabled": true
    },
    ...
  ]
}
```

- Allows API hosts to configure “Ownership-Based” Authorization Strategy for any resource/action.
- API authorizes access to individual resources based on the concept of “ownership”, when configured.
EdFi_Ods

Each aggregate root table in the ODS has `CreatedByOwnershipTokenId` column (smallint).
EdFi_Admin

Tokens currently "owned" by the API client (can be used in ownership transfer scenarios)

Ownership token associated with the API client for creation
Implementation

- API Clients are associated with Ownership tokens
- For each record **insert** request
  - Follows the existing authorization strategy
  - Sets the `CreatedByOwnershipTokenId` property of the root aggregate to the `CreatorOwnershipTokenId` of the API Client
- For each record **read, update or delete** request, if ownership-based authorization strategy is set for the resource/action
  - Compares the `CreatedByOwnershipTokenId` from the table record to the `ApiClient's OwnershipTokens`. If they are the same then allow the request to pass through.
  - If they are not the same, then reject the request and throw an unauthorized error
Support For External Cache
Support For External Cache

- Optional External Cache in place of “in-memory cache”
- Optimal memory usage on containerized applications that scale up based on utilization
- Avoids cache sync problems with unified cache
- Caution: Use of an external cache will introduce some level of latency into the response times compared to in-memory cache
Technology Upgrade
ODS API v5.3

- .NET Core 3.1 (end of support in December 2022).
- SQL Server 2016
- PostgreSQL 11

ODS API v6.0

- .NET 6 with LTS support and performance improvements.
- SQL Server 2019
- PostgreSQL 13