What's New in v3.0 - Feature & Change Summary

This section provides an overview of the new features in the v3.0 Release Candidate of the Ed-Fi ODS / API. All features below are currently expected to be in the final release. A comprehensive listing of all changes is documented in the Release Notes section.

Contents

Find out more about what's new in the latest version of the Ed-Fi ODS / API:

The major changes include:

- New & Updated Standards
  - Alignment With the Upcoming Ed-Fi Data Standard v3.0
  - Upgrade to OpenAPI 2.0 Specification
  - Standardized Use of OAuth 2.0 Client Credentials Grant Type
- New & Improved Features
  - Decoupling of Ed-Fi Core and Extensions in the API
  - Improvements to Routes
  - Support for Multiple Operational Contexts
- Code Refactoring

Detail about each change follows.

New & Updated Standards

Several changes were made to the ODS / API v3.0 to update and adhere to standards.

Alignment With the Upcoming Ed-Fi Data Standard v3.0

Every generation of Ed-Fi technology is aligned with the Ed-Fi Data Standard. The ODS / API v3.0 Release Candidate implements the Data Standard v3.0d, which is a draft that contains improvements and additions based on feedback from implementers. The final ODS / API v3.0 will align with the final Data Standard v3.0.

- The changes in the latest Data Standard have an impact to the surface of the ODS / API. See the latest Ed-Fi Data Standard v3.0 documentation for more details.
- API documentation for the v3.0 ODS / API Release Candidate is available online at https://api.ed-fi.org/v3/docs/.
- Maintenance releases of the Ed-Fi ODS / API v2.x platform will continue to align with Ed-Fi Data Standard v2.0.

Upgrade to OpenAPI 2.0 Specification

This change upgrades the Ed-Fi ODS / API to use the Swagger / OpenAPI specification 2.0.

The previous Ed-Fi ODS / API v2.x produced Swagger 1.2 metadata. This specification is out-of-date, and tooling support is lacking going forward – which means that the SwaggerUI can no longer be updated without an upgrade to the metadata.

Overview

The upgrade to OpenAPI 2.0 brings the following benefits:

- Moves to an actively supported and maintained version of Swagger.
- Adopts Swagger community tools for generating client SDKs for the ODS / API. This provides API SDK support for languages beyond C# and Java, as well as enables the Alliance to deprecate costly support and maintenance for the current Client API SDK.
- Standardizes the return type for HTTP GETs.
- Enables significantly improved runtime performance of the latest Swagger UI.

Code Generation

The Swagger Codegen tooling is a template-driven engine that can generate documentation, API clients, and server stubs in different languages by parsing an OpenAPI specification. Client languages include ActionScript, Bash, C#, C++, Erlang, Go, Groovy, Haskell, Java, Node.js, Objective-C, Perl, PHP, PowerShell, Python, R, Ruby, Scala, Swift, and others.

The free, open-source Swagger Codegen project is maintained in GitHub at https://github.com/swagger-api/swagger-codegen. Any version of Swagger Codegen compatible with the OpenAPI 2.0 specification can generate client code for the Ed-Fi ODS / API v3.0.

Design Details

The metadata exposed by the Ed-Fi ODS / API is used for two purposes. One is to allow interactive exploration of the API using SwaggerUI. The other is to generate SDK client code. With these use cases in mind, the API metadata is being reorganized around a set of distinct metadata endpoints.

- As with the current implementation, there will be API sections for Resources, Descriptors, Bulk, Identity, and one for each Profiles and Composites.
- ODS / API v3.0 introduces new metadata endpoints specifically targeted for code generation.

The table below summarizes these endpoints.

Name	Description	Primary Target Audience
Resources	Includes all the data collection resources including Extensions (but excludes the Descriptors).	API consumers and application vendors using SwaggerUI.
Descriptors	Includes only the Ed-Fi Descriptors (including Descriptors added via Extensions).	API consumers and application vendors using SwaggerUI.
Bulk	Contains all operations related to bulk data loads.	API consumers and application vendors who are operating against an API that supports bulk loads.
Identity	Contains all operations supporting integration with an external UniqueId system.	API consumers and application vendors working with APIs that have an integration with an external UniqueId system.
Profile-Specific	Includes the relevant subset of the Standard API Resources (including extensions and any referenced Descriptors) with appropriately constrained surface areas.	Specialty vendors, for data collection and/or integration.
Composite-Specific	Includes read-only resources, as defined in the composite, including any referenced descriptors.	API consumers and application vendors focused on data integration activities in a specific subject area.
Data Collection API	Includes all Descriptors and Resources, including all Extensions.	SDK generators.

Standardized Use of OAuth 2.0 Client Credentials Grant Type

The ODS / API v2.x authentication process used a customized approach that looked like "2.5 legged OAuth," meaning that it leveraged idioms from both 2-legged and 3-legged OAuth, but followed neither specification exactly. Since the API authenticates systems (not users) and treats those systems as the “resource owner,” no “authorization code” is necessary, but was nevertheless required in the v2.x authorization flow.

Rather than continue this custom implementation, the ODS / API v3.0 implements the OAuth 2.0 Client Credentials Grant Flow. This enables the ODS / API v3.0 to work "out-of-the-box" with standard tools and SDKs.

The API authentication process no longer uses “authorization codes," and thus all technical artifacts related to the concept have been removed. This necessitated changes to the Admin database by removing a table (dbo.ClientAuthorizationCodes), updating stored procedures (e.g., AccessTokenIsValid), as well as various C# classes (e.g., the OAuthController and its associated artifacts). New tests have been written to cover the new functionality, and obsolete tests have been removed.

Additionally, all behavior related to the rolling expiration functionality of the existing tokens has been removed to comply with the OAuth 2.0 standards.

New & Improved Features

This section summarizes the major new features of the ODS / API v3.0.

Decoupling of Ed-Fi Core and Extensions in the API

The goal of this feature is to completely decouple the API artifacts for Ed-Fi Core from those for Extensions. The draft articles How To: Extend the Ed-Fi ODS / API - Student Transcript Example and How To: Extend the Ed-Fi ODS / API - Student Transportation Example provide detailed instructions.

This change allows implementations to support Ed-Fi Core data elements and add Ed-Fi Extensions based on a given configuration. It will also be possible to apply multiple Extensions simultaneously, such as host-specific Extensions (e.g., as implemented by many State Education Agencies), along with domain-specific extensions (e.g., talent management or transportation). The issues and limitations related to the current ODS / API support for Extensions are documented here.

Support for Extensions in JSON Resource Models

Instead of a seamless blending of extended elements into the JSON payload, extended elements of an existing entity will be held in an extension JSON object. Current (ODS / API v2.x) and future (ODS / API v3.0) examples follow.

Current (ODS / API v2.x)	Future: (ODS / API v3.0)

{
    "studentUniqueId": "397589871",
    "firstName": "John",
    "lastSurname": "Ortiz",
    "tribalAffiliation": "Pascua Yaqui"
 	
    [etc...]
 	
}
	{
	    "studentUniqueId": "397589871",
	    "firstName" : "John",
	    "lastSurname:" "Ortiz",
	
	[etc...]
	
	    “_ext” : {
	        “SomeState” : {
	            “tribalAffiliation” : “Pascua Yaqui”,
	            [etc...]
	        }
	    }
	}

The Improvements to Routes section that follows includes an overview on how these changes are reflected in the URI routes.

MetaEd IDE Extension Support

The Ed-Fi ODS / API can easily be extended using the MetaEd IDE. MetaEd v1.1.5 provides support for the ODS / API v3.0 Release Candidate. See MetaEd IDE - What's New for details.

XML Shredder Support for Decoupled Extensions

This feature also includes XML Shredder Resource Factory enhancements for resource model changes.

Please note that the bulk loading is being finalized as of the Release Candidate. See ODS-1750 for further details.

Improvements to Routes

The ODS / API v3.0 includes improvements to the URI routes that clients use to connect to the platform.

Improvement Summary

The following changes will be made to the URI paths, to both support the new extensions model, and provide improved support for API versioning as shown.

Description	Current (ODS / API v2.0)	Future (ODS / API v3.0)
Base URI	../api/v2.0/2018	/{api_area}/{version}/{year}* **Examples:** /data/v3 /composites/v1 /identity/v2 /bulk/v1

Basic Relative URL Format		Data Management Resources:
		…/{org}/{resource}
		
		API Composites:
		.../{org}/{category}/{resource}

Examples		

Resources (Ed-Fi)	/api/v2.0/2018/schools/	/data/v3/ed-fi/schools
Descriptors (Ed-Fi)	/api/v2.0/2018/termDescriptors	/data/v3/ed-fi/termDescriptors
Resources (Host Extensions)	/api/v2.0/2018/leavers	/data/v3/TX/leavers
Resources (Domain Extensions)	/api/v2.0/2018/applicants	/data/v3/talentMgmt/applicants
Composites (Ed-Fi)	/api/v2.0/2018/assessment /studentAssessments	/composites/v1/ed-fi/assessment /studentAssessments
Composites (Host Organization)	Custom	/composites/v1/MI/enrollment /Schools
Composites Based on Domain or Host Extensions	N/A	Unsupported
Bulk	api/v2.0/2018/bulkOperations	/bulk/v1/bulkOperations
Identity	api/identity/v1/identities api/identity/v2/identities	/identity/v2/identities

Notes:

> * School Year is now optional in the route
> ** Composites are treated as a reserved word
> *** “Org” name registration will be managed by the Ed-Fi Alliance (or self-managed by community through conventions for SEAs, LEAs)

Separate Bulk and Identity Endpoints

In addition to the general improvements noted above, the endpoints for bulk and identity features are now clearly separated. This allows separate versioning for these features, and also better supports implementations that are not using the optional identity and bulk features. Examples include:

- /data/v3
- /identity/v2
- /bulk/v1

Configurable School Year

Some implementers rely on including the school year in the API root, while others do not. The presence of the school year value in the route is now configurable, and off by default.

> School year can be configured in the API route through the following steps:
>
> - Update owin:appStartup key in the Web.config of the EdFi.Ods.WebApi project to: <add key="owin:appStartup" value="YearSpecific" />
> - Update swagger.webApiMetadataUrl key in the Web.config of the EdFi.Ods.SwaggerUI project to: <add key="swagger.webApiMetadataUrl" value="http://localhost:54746/metadata/2017/"/>

Support for Multiple Operational Contexts

The ODS / API v3.0 sets the stage to support multiple operational contexts.

Ed-Fi Descriptor References Require Fully Qualified Namespaces

The Ed-Fi Descriptor pattern allows controlled vocabularies to be customized for the needs of a particular context. Many commonly used option lists such as Academic Subject, Grading Period, Language, Term, and so forth are defined by Descriptors.

All Descriptor references now require namespaces, and do not rely on the concept of a default operational context. This change ensures that the semantics of all messages are clear, even in transit. This change also communicates information about who governs the values being sent for the controlled vocabularies that make up a particular Descriptor.

The Descriptor references must be formatted as follows:

```
uri://[organization indicator]/[name of descriptor]#[code value]
```

The following are example from the Ed-Fi Alliance namespace, which includes the default values in the as-shipped ODS / API:

```
uri://ed-fi.org/AcademicSubjectDescriptor#Chemistry
uri://ed-fi.org/GradeLevelDescriptor#Fifth grade
```

Namespaces are assigned by platform hosts. Generally speaking, the namespace should clearly identify the originating entity in the `[organization indicator]` segment. In addition, the namespace MUST adhere to the following patterns:

- Valid namespace format is: [scheme]://[organization indicator]/[descriptor name]
- Valid scheme is always 'uri'
- Valid organization indicator can only contain alphanumeric and $-_.+!*'(),
- Valid descriptor name can only contain alphanumeric
- Valid code value can contain anything except '#'

The ODS / API v3.0 accepts the new format and provides validation errors for data in the incorrect format.

Code Refactoring

The ODS / API v3.0 includes a refactored code base for easier understanding and maintenance.

- The source code reorganization addresses complexity and establishes a foundation for more manageable, composable releases. For example, the refactoring will support distribution of the ODS / API core via NuGet packages in a future release.
- A reengineering of the API code generation templates addresses technical debt, and prepares for API code generation support using the Ed-Fi METAED software.

Note that these code refactoring changes for ODS / API v3.0 were also applied to the ODS / API v2.3 release.