

# Transition to 3.0 Session Notes

Thursday, April 12, 2018

9:30 AM - 11:30 AM	Transition to 3.0 ODS and API, including using the new extensibility model
	Ballroom AB
	This in-depth session will help prepare agencies and vendors for transition to the 3.0 APIs and the new model for extensibility.



These broad session notes attempt to capture the spirit of the discussion and should not be interpreted as a transcript. Although Ed-Fi Alliance staff were involved in capturing these observations, the notes below should not be construed as official, complete, or 100% accurate.

Presentation by [Vinaya Mayya](#)

The slide deck will be posted. These notes will just cover highlights and discussion.

Key references:

- [ED-FL RFC 12 - DATA STANDARD v3.0d](#)
- [Ed-Fi ODS / API v3.0](#)

## Expansion of Support for Standardized Federal EDFacts Data Collection

- Four new StudentProgramAssociations - do these now cover everything for EdFacts?
  - No. It does bring us closer, but not full coverage yet. Extensibility model will allow you to cover the gaps as needed.
  - If you already have extensions for these four, then you'll want to remove them.

There are more changes that are not covered in this presentation, such as changes around *Section*. The 2:15 session will go through more details.

## Upgrade to OpenAPI 2.0 Specification

- Using the standard, current, Swagger now instead of the outdated and highly customized swagger-ui used for ODS 2.x.
- Big performance improvements, and useful UI improvements

## Standardized GET calls

- Essentially, only have GetByExample now - whether querying by natural key or by example, you will always know that a collection will be returned (even if it is just an empty array).
- How do you know what the natural key is? Look in Swagger. The metadata now describes which fields are part of the key ("is required").

[blocked URL](#)

- GetById still returns single item.

Any vendor requests for delete by natural key? under discussion

## Authentication

- ODS / API 2.x used [3-legged OAuth](#) ... almost. But slightly different because we don't have users, only client applications. Result: chatty, with unnecessary calls.
- ODS / API 3.0 using [2-legged OAuth](#) - client resource grant type. Greatly simplifies the data flow and reduces the number of network calls.

## Extensions & Core

- Wanted to separate out extensions from core data in JSON so that a developer can tell that the data is an extension.
  - { ... "\_ext": { "namespace": { "someProperty": ... } } }
- Route has the extension in it: /data/v3/<namespace>/<resource>
- Who controls the namespaces?
  - Ed-Fi currently not registering custom namespaces
  - MetaEd developer will define the namespace
- There is a distinction between extending an entity ( "\_ext" shown above) and an extension entity (corresponds to the route mechanism shown above)
  - Can we call it a "resource extension" instead of an "extension entity"?
    - That term is already used for aggregate extensions

- Some concern about how vendors will handle this because two different implementations might re-use the same schema names for different purposes
  - Current discussion is to use state code (for SEA) for namespace, e.g. "AZ"
    - Request to make sure this is documented in Tech Docs ([Vinaya Mayya?](#))
  - No convention yet on domain extensions
- What about versioning?
  - Custom extensions can be versioned on the route
  - "v3" in the route is for the overall technology, not the extensions themselves
  - Extenders should use versions in their routes to communicate to others if the interface changes
  - Concern around maintaining multiple versions of extensions arises from state mandates to be able to access / modify 3 years of data - hence backwards compatibility required

Is there any way to search for extensions that others have already created?

- MappingEdu?
- Is there any way to easily compare the changes between two versions?

MetaEd demonstration

- Tech Docs "getting started" will walk you through similar steps
- New deployment command simplifies process of taking MetaEd output and incorporating into the ODS/API
- This version of MetaEd supports both 2.x and 3.0. Must use "Version Targeting" to set version of ODS.
- Even if not working on 3.0 implementation yet, use MetaEd now for best practice

How do you ensure that the API works?

- Smoke test suite
- How robust? Focused on non-destructive tests first. After that creates a sandbox and runs destructive tests.
- Doesn't support extensions (yet). You would need to load your own non-destructive data to prime the GET requests.
- Around 2400 unit and integration tests running for each check-in

3.0 standard expected to be locked-down late May.

- Concern that this time frame isn't very good for school districts who are closing out the year end. will continue discussing whether it makes sense to hold in Release Candidate status for a longer period of time.
- Will still be able to make minor updates, so long as they aren't breaking changes.