

# Unique ID System Integration

Generally speaking, a unique ID system ensures that a person has only one unique ID for an organization even if the person is represented in multiple roles, such as an individual who is both a staff member and parent at a school. The Ed-Fi ODS / API provides a set of endpoints to create and retrieve unique IDs for person records including students, staff, and parents.

Many organizations have an existing implementation of a unique ID system. The Ed-Fi ODS / API does not replace an organization's unique ID system; instead it provides a standard interface for use by clients and clear extension points for platform developers to integrate with their existing system.

The Ed-Fi ODS / API ships with its unique ID endpoints configured to return a `501 - Not Implemented` response. Integration work is required to connect to an enterprise unique ID system. This documentation contains conceptual and how-to material about connecting to an existing unique ID system.

## Unique ID System Integration Models

Before unique ID system integration begins, platform hosts must select a platform model that best serves their enterprise's needs. There are two distinct models: an integrated unique ID system model (shortened to "integrated model," below), and a non-integrated unique ID system model ("non-integrated," below).

### Integrated Model

This model is suitable when an external system, often a commercial product, is the authoritative source for unique IDs and can be wired into the ODS / API. In this model, an external unique ID system always supplies the unique ID for students, staff, and parent entities.

Distinguishing characteristics of this model include:

- Unique IDs are established outside the context of the ODS / API by the external unique ID system.
- Client applications use the API identities resource to look up student records and request new unique IDs.
- When new person entities are created, the external system automatically assigns a new unique ID. Client applications do not set unique ID values.

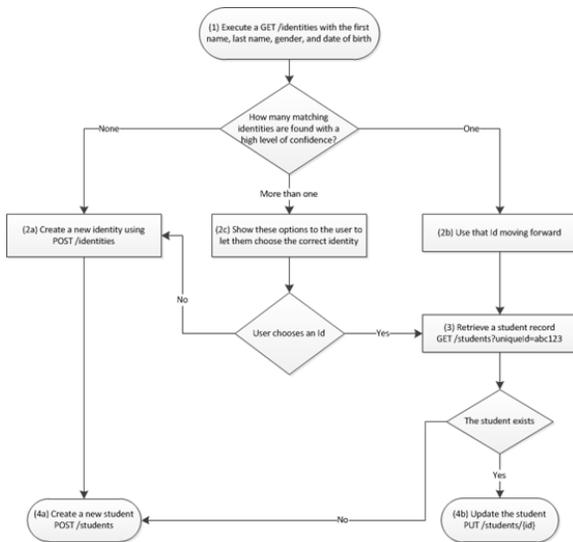
This model automates the assignment of unique IDs, but does make some assumptions about the system flow. An outline of the usual system flow follows.

#### Client-side ID Lookup in the Integrated Model

When the ODS / API platform uses an integrated model for establishing a unique ID, client applications will need some process for looking up that unique ID value. This section provides a system flow that most clients will follow to look up a unique ID from the Ed-Fi ODS / API.

The unique ID for a person can be obtained from the Ed-Fi ODS / API UniqueId endpoints. The workflow for clients to create and obtain IDs are described below. For the purposes of this example, the workflow will focus on the use case of creating a new student record, but the same general flow applies to staff and parent records.

It is assumed that once a Unique ID is obtained by a client, it will be stored in client applications as a way to easily sync data between the client and platform from that point forward.



A description of each step follows.

- **Step 1.** Execute a GET /identities with the first name, last name, gender, and date of birth.
- **Step 2a.** No identities are returned, so create a new identity using POST /identities
- **Step 2b.** A single identity is returned, so use that one as a pre-existing Id.
- **Step 2c.** Multiple identities are returned. Show these options to the user to let them choose. If they choose one, then use that as the pre-existing identity. If they choose none of the above, then create a new identity using POST /identities
- **Step 3.** If an identity already existed for the student. Execute a GET /students?uniqueid=xxxx to obtain the existing student record.
- **Step 4b.** A student record does not exist. Create a new student using a POST /students
- **Step 4a.** A student record exists. Modify the student record and execute a PUT /students/{id} to update the entity.

Now that the Unique ID has been obtained, it can be used by clients in bulk and transactional operations when referring to that student.



When a client application creates a student record, the client won't be able to retrieve or update that student until it has a security context. That context is created by adding a StudentSchoolAssociation record under an LEA to which the client application has access. Similarly, when a client application creates a staff record, the client won't be able to retrieve or update that staff record until it adds a staffEducationOrganizationAssignmentAssociation or staffEducationOrganizationEmploymentAssociation under an LEA to which the client application has access.

## Non-integrated Unique ID System Model

This model is suitable when platform hosts choose not to wire in or link to an external unique Id system. In this model, platform hosts provide guidance to clients on how to acquire or assign a unique ID out of band. One approach is to require clients supply a GUID when creating new person entities. Another approach is to have clients acquire an ID from an enterprise system and pass that value to the ODS / API when creating new person entities.

Distinguishing characteristics of this model include:

- An approach to creating or assigning unique IDs is determined by the platform host.
- Client applications are responsible for populating the unique ID for person entities.

This model requires less system integration from a platform host, but requires that reasonably specific guidance be provided to client application developers.

## Integration Model Comparison

The above models are fundamentally different for the platform host, but from a client's perspective the change is limited to whether or not the client application supplies the unique ID – and if the client is responsible for assigning the ID, exactly how that value should be determined.

## Key Integration Steps

Regardless of whether you choose an integrated or non-integrated model, many of the key points are the same.

## Implementing IUniqueIdentity

Integrating your unique ID system with the Ed-Fi ODS / API requires a custom implementation of the IUniqueIdentity interface to be written. When properly registered, this code is called for each of the unique ID API calls received by the Ed-Fi ODS / API.

The EdFi.Common project contains the definition for the IUniqueIdentity and IIdentity interfaces. The placeholder implementation of this interface is located in the EdFi.Identity project under the Models /UniqueIdentity.cs code file.

```

Unique Identity Interfaces

public interface IUniqueIdentity
{
    IIdentity Get(string uniqueId);
    IIdentity[] Get(IIdentity identity);
    IIdentity Post(IIdentity command);
}

public interface IIdentity
{
    string UniqueId { get; set; }
    Gender? BirthGender { get; set; }
    DateTime? BirthDate { get; set; }
    string FamilyNames { get; set; }
    string GivenNames { get; set; }
    double Weight { get; set; }
    bool IsMatch { get; set; }
    IIdentifier[] Identifiers { get; set; }
}

```

Member	Description
uniqueId	The human-readable unique identifier for the person. While the uniqueId is represented as a string in the interface, the Ed-Fi ODS Database limits the length of the uniqueId to 32 characters.
FamilyNames	A space-delimited list of all the family names associated with a person. In some Hispanic and other traditions this may be both parents' surnames (e.g., Hernandez Gonzalez). In some Asian and other traditions, the surname may be provided first for each child.
GivenNames	A space-delimited list of all the names given to the person. Typically first and middle names, but also suffixes (e.g., Jr., III, Esquire)
Weight	A number indicating the level of confidence with the provided match. Higher numbers are more confident. Some systems use the number of standard deviations from normal (sigma) and others use a 0 to 100 scale. In other systems, the number is relative to the result set provided and has no relevance outside of one query. This is an implementation-specific detail that would be relayed to a human user of the system for resolution.
BirthGender	The natural gender of the person at the time of their birth.
Identifiers	An optional list of third-party identifiers (e.g., local education organization identifiers, driver's license, teaching certificate number) that could be used to further distinguish a person. These additional identifiers should only be stored if they are used as a matching criteria by the native Unique Identity system.

## Register the Unique Identity System

The WebApiInstaller class (in the EdFi.Ods.Api project) registers the placeholder UniqueIdentity class in its RegisterEduIdDependencies method. This installer is called by the OWIN startup class referenced in the EdFi.Ods.WebApi web.config file. Use this pattern when registering your unique identity implementation in your custom OWIN startup class.

### UniqueIdentity Registration Code

```
// Need to register the API endpoint
container.Register(
    Classes.FromAssemblyContaining<Marker_EdFi_Identity>()
        .BasedOn<ApiController>()
        .LifestyleScoped());

// Now register your UniqueIdentity implementation
container.Register(Component
    .For<IUniqueIdentity>()
    .ImplementedBy<My.Custom.UniqueIdentityImplementation>());
```