

How To: Extend the Ed-Fi ODS / API - Student Transportation Example

In this example, we will create a new domain entity called StudentTransportation. This entity will be an extension to the Student Enrollment interchange and will be exposed in Ed-Fi ODS / API through a new API resource called **studentTransportations**. It is assumed that the Ed-Fi ODS has been successfully downloaded and is running as in a local environment per the instructions in the [Getting Started](#) document ation.

The steps can be summarized as:

- [Step 1. Author Ed-Fi Core Schema Extensions](#)
- [Step 2. Author a Custom Interchange Schema](#)
- [Step 3. Author Database Schema Extensions](#)
- [Step 4. Author API Metadata Extensions](#)
- [Step 5. Run Code Generation and Verify Changes](#)

Each step is outlined in detail, below.

Step 1. Author Ed-Fi Core Schema Extensions

Create an extension to the Ed-Fi Core Schema, called **EXTENSION-Ed-Fi-Core.xsd**, and place it in the **C:\Ed-Fi-ODS-Implementation\Extensions\Schemas** folder. This extension introduces new entity types for the StudentTransportation entity. It is important to note that core schema extension file must be able to resolve the reference to the Ed-Fi Core Schema file.

EXTENSION-Ed-Fi-Core.xsd

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns="http://ed-fi.org/0200" xmlns:ann="http://ed-fi.org/annotation" xmlns:altova="http://www.altova.com/xml-schema-extensions" targetNamespace="http://ed-fi.org/0200" elementFormDefault="qualified" attributeFormDefault="unqualified">
  <xs:include schemaLocation="..\..\..\Ed-Fi-ODS\Application\EdFi.Ods.CodeGen\App_Packages\Ed-Fi\Schema\Ed-Fi-Core.xsd"/>
  <xs:annotation>
    <xs:documentation>==== Sample Core Type Extensions ====</xs:documentation>
  </xs:annotation>
  <xs:complexType name="EXTENSION-StudentTransportation">
    <xs:annotation>
      <xs:documentation>New state-specific entity which specifies the buses that a student is expecting to use and the approximate distance.</xs:documentation>
    </xs:annotation>
    <xs:appinfo>
      <ann:TypeGroup>Domain Entity</ann:TypeGroup>
    </xs:appinfo>
    </xs:annotation>
    <xs:complexContent>
      <xs:extension base="ComplexObjectType">
        <xs:sequence>
          <xs:element name="StudentReference" type="StudentReferenceType">
            <xs:annotation>
              <xs:documentation>Student associated with the buses.</xs:documentation>
            </xs:annotation>
          </xs:element>
          <xs:element name="SchoolReference" type="SchoolReferenceType">
            <xs:annotation>
              <xs:documentation>School associated with the buses.</xs:documentation>
            </xs:annotation>
          </xs:element>
          <xs:element name="AMBusNumber" type="EXTENSION-BusNumber">
            <xs:annotation>
```

Downloads

The following link is a ZIP archive containing source files for this extensibility sample.

[Student Transportation Source Files](#)

```

                <xs:documentation>Bus number student rides in
morning (AM)</xs:documentation>
            </xs:annotation>
        </xs:element>
        <xs:element name="PMBusNumber" type="EXTENSION-
BusNumber">
            <xs:annotation>
                <xs:documentation>Bus number student rides in
afternoon (PM)</xs:documentation>
            </xs:annotation>
        </xs:element>
        <xs:element name="EstimatedMilesFromSchool" type="
EXTENSION-EstimatedMiles">
            <xs:annotation>
                <xs:documentation>The distance the child lives
from the school. Example, 1 Mile = 01.00 2.5 Miles = 02.50</xs:
documentation>
            </xs:annotation>
        </xs:element>
    </xs:sequence>
</xs:extension>
</xs:complexContent>
</xs:complexType>
<xs:simpleType name="EXTENSION-BusNumber">
    <xs:annotation>
        <xs:documentation>Bus number student rides (see EXTENSION-
StudentTransportation - used for AM and PM bus numbers)</xs:documentation>
    <xs:appinfo>
        <ann:TypeGroup>Simple</ann:TypeGroup>
    </xs:appinfo>
</xs:annotation>
    <xs:restriction base="xs:string">
        <xs:minLength value="0"/>
        <xs:maxLength value="6"/>
    </xs:restriction>
</xs:simpleType>
<xs:simpleType name="EXTENSION-EstimatedMiles">
    <xs:annotation>
        <xs:documentation>Estimated distance from point A to point B.
(see EXTENSION-StudentTransportation)</xs:documentation>
    <xs:appinfo>
        <ann:TypeGroup>Simple</ann:TypeGroup>
    </xs:appinfo>
</xs:annotation>
    <xs:restriction base="xs:decimal">
        <xs:minInclusive value="0.00"/>
        <xs:maxInclusive value="999.00"/>
        <xs:totalDigits value="5"/>
        <xs:fractionDigits value="2"/>
    </xs:restriction>
</xs:simpleType>
</xs:schema>

```



For more information about how to extend the Ed-Fi Core Schema, see [XML Schema - Extensions Framework Guide](#).

Step 2. Author a Custom Interchange Schema

Create an custom interchange file, called **EXTENSION-Interchange-StudentEnrollment.xsd**, and place it in the **C:\Ed-Fi-ODS-Implementation\Extensions\Schemas** folder. This file overrides the Ed-Fi [Student Enrollment Interchange](#), adding in the StudentTransportation entity into the Interchange. It is important to note that the `schemaLocation` should be a valid reference the schema file that contains the extension definition (i.e., the `EXTENSION-Ed-Fi-Core.xsd` file created in the previous step).

EXTENSION-Interchange-StudentEnrollment.xsd

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns="http://ed-fi.
org/0200" targetNamespace="http://ed-fi.org/0200" elementFormDefault="
qualified" attributeFormDefault="unqualified">
  <xs:include schemaLocation=".\\EXTENSION-Ed-Fi-Core.xsd"/>
  <xs:annotation>
    <xs:documentation>==== Student Enrollment Interchange Model
====</xs:documentation>
  </xs:annotation>
  <xs:element name="InterchangeStudentEnrollment">
    <xs:annotation>
      <xs:documentation>The Student Enrollment interchange describes
student enrollments in schools and in sections.</xs:documentation>
    </xs:annotation>
    <xs:complexType>
      <xs:choice maxOccurs="unbounded">
        <xs:element name="SectionReference" type="SectionReferenceType"
/>
        <xs:element name="StudentSchoolAssociation" type="
StudentSchoolAssociation"/>
        <xs:element name="StudentSectionAssociation" type="
StudentSectionAssociation"/>
        <xs:element name="GraduationPlan" type="GraduationPlan"/>
        <xs:element name="StudentEducationOrganizationAssociation"
type="StudentEducationOrganizationAssociation"/>
        <xs:element name="StudentTransportation" type="EXTENSION-
StudentTransportation"/>
      </xs:choice>
    </xs:complexType>
  </xs:element>
</xs:schema>
```



For more information about how to create a custom interchange schema, see [XML Schema - Custom Interchange Schema](#).

Step 3. Author Database Schema Extensions

Create an extension SQL script called **0001-Extensions.sql** and place it in the **C:\Ed-Fi-ODS-Implementation\Database\Structure\EdFi** folder. This script defines the database schema for the extension.

0001-Extensions.sql

```
IF NOT EXISTS (SELECT 1 FROM sys.schemas WHERE name = 'extension')
BEGIN
    EXEC('CREATE SCHEMA [extension] AUTHORIZATION dbo')
END
GO

IF OBJECT_ID('extension.FK_StudentTransportation_SchoolId') IS NOT NULL
    ALTER TABLE extension.StudentTransportation DROP CONSTRAINT
    FK_StudentTransportation_SchoolId;

IF OBJECT_ID('extension.FK_StudentTransportation_StudentUSI') IS NOT NULL
    ALTER TABLE extension.StudentTransportation DROP CONSTRAINT
    FK_StudentTransportation_StudentUSI;

IF EXISTS (SELECT 1 FROM sys.tables WHERE SCHEMA_NAME(schema_id) =
'extension' AND OBJECT_NAME(object_id) = 'StudentTransportation')
    DROP TABLE extension.StudentTransportation;

IF NOT EXISTS (SELECT 1 FROM sys.tables WHERE SCHEMA_NAME(schema_id) =
'extension' AND OBJECT_NAME(object_id) = 'StudentTransportation')
BEGIN
    CREATE TABLE extension.StudentTransportation (
        Id uniqueidentifier NOT NULL CONSTRAINT
        StudentTransportation_DF_Id DEFAULT newid()
        ,SchoolId INT NOT NULL
        ,StudentUSI INT NOT NULL
        ,AMBusNumber VARCHAR(6) CONSTRAINT
        StudentTransportation_DF_AMBusNumber DEFAULT '-' NOT NULL
        ,PMBusNumber VARCHAR(6) CONSTRAINT
        StudentTransportation_DF_PMBusNumber DEFAULT '-' NOT NULL
        ,EstimatedMilesFromSchool DECIMAL(5,2) NOT NULL
        ,CreateDate datetime NOT NULL CONSTRAINT
        StudentTransportation_DF_CreateDate DEFAULT getdate()
        ,LastModifiedDate datetime NOT NULL CONSTRAINT
        StudentTransportation_DF_LastModifiedDate DEFAULT getdate()
        ,CONSTRAINT StudentTransportation_PK PRIMARY KEY CLUSTERED
        (SchoolId,StudentUSI,AMBusNumber,PMBusNumber)
        ,CONSTRAINT FK_StudentTransportation_SchoolId FOREIGN KEY
        (SchoolId) REFERENCES edfi.School (SchoolId)
        ,CONSTRAINT FK_StudentTransportation_StudentUSI FOREIGN KEY
        (StudentUSI) REFERENCES edfi.Student (StudentUSI)
    )
    CREATE UNIQUE NONCLUSTERED INDEX GUID_StudentTransportation ON
    extension.StudentTransportation (Id);
END
GO

EXEC sys.sp_addextendedproperty 'MS_Description', 'A designation of the
transportation a student uses to get to and from school.', 'schema',
'extension', 'table', 'StudentTransportation'
GO
```



When modeling the extension database tables, it is important to follow the patterns that already exist in the database.

Step 4. Author API Metadata Extensions

Create an extension API metadata file called **DomainMetadata-Extension.xml** and place it in the **C:\Ed-Fi-ODS-Implementation\Extensions\Metadata** folder. This file will register the StudentTransportation entity as a domain aggregate and ensure the code generation process will process the entity.

DomainMetadata-Extension.xml

```
<?xml version="1.0" encoding="utf-8" ?>
<AggregateExtensions>
  <Aggregate root="StudentTransportation">
    <Entity table="StudentTransportation" schema="extension" />
  </Aggregate>
</AggregateExtensions>
```

Step 5. Run Code Generation and Verify Changes

Re-run the code generation steps outlined in the Getting Started Guide, (i.e., from a PowerShell prompt run `Initialize-PowershellForDevelopment.ps` script, followed by the `initdev` command). Then run the application and view the Ed-Fi ODS / API using Swagger. The following new API resource should be visible:

studentTransportations :		Show/Hide	List Operations	Expand Operations	Raw
GET	/studentTransportations		Retrieves resources based with paging capabilities (using the "Get All" pattern).		
GET	/studentTransportations		Retrieves a specific resource using the values of the resource's natural key (using the "Get By Key" pattern).		
POST	/studentTransportations		Creates or updates resources based on the natural key values of the supplied resource.		
GET	/studentTransportations/{id}		Retrieves a specific resource using the resource's identifier (using the "Get By Id" pattern).		
PUT	/studentTransportations/{id}		Updates an existing resource based on the resource identifier.		
DELETE	/studentTransportations/{id}		Deletes an existing resource using the resource identifier.		