

# API SDK for C#

This section outlines how to use code generation to create an Ed-Fi ODS / API Client SDK using a Windows environment targeting C#. The high-level steps are:

- [Step 1. Install and Configure Required Software](#)
- [Step 2. Download the Ed-Fi ODS Client SDK Source Code](#)
- [Step 3. Build the Swagger Codegen JAR File](#)
- [Step 4. Generate the SDK Source Files](#)
- [Step 5. Use the SDK in a Sample C# Program](#)

Each step is outlined in detail below.

## Step 1. Install and Configure Required Software

This step ensures you have the prerequisite software on your development machine. In case you're wondering: the code generation leverages Java, but it does output C# code.

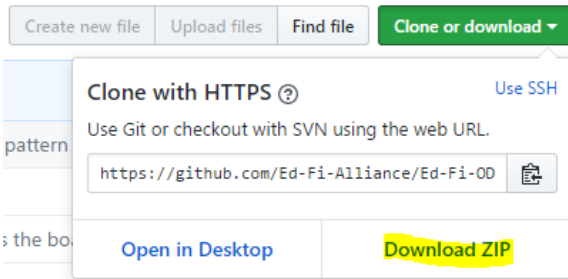
- Install the [Java JDK Version 8](#).
- Install the [Scala Build Tool](#) (sbt).
- Set the Windows PATH environment variable to find the jdk bin directory.

## Step 2. Download the Ed-Fi ODS Client SDK Source Code

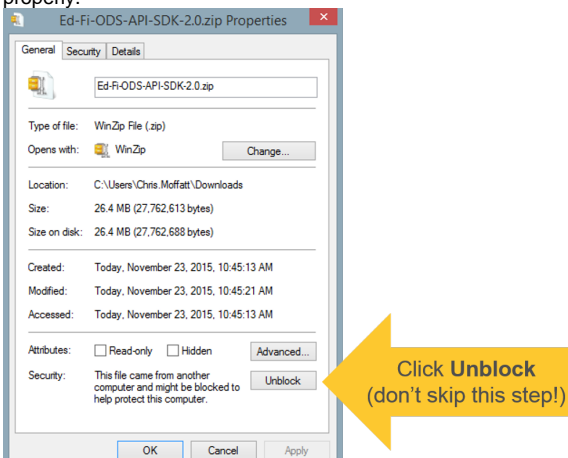
The Ed-Fi ODS / API Client SDK source code is contained in an Ed-Fi repository called Ed-Fi-ODS-API-SDK.

Follow these steps to download the repository archive and extract it:

- Navigate to GitHub and select the appropriate version of the source code:
  - Latest Stable Release (v2.4): <https://github.com/Ed-Fi-Alliance/Ed-Fi-ODS-API-SDK/tree/v2.4.0>
  - Latest Daily Source: <https://github.com/Ed-Fi-Alliance/Ed-Fi-ODS-API-SDK/tree/development>
- Download the repository archive to your local drive.



- In Windows Explorer, right-click on the downloaded ZIP files and select "Properties." On the General tab, press the **Unblock** button to allow the contents of the contained scripts to execute properly.



## API Client Developers' Guide Contents

Find out more about how to develop client applications for an Ed-Fi ODS / API v2.4 platform:

### Downloads

The following link is a ZIP archive containing a C# sample program that uses the client SDK:

#### [C# SDK Sample App](#)

The Sample program works against the Ed-Fi ODS / API sandbox hosted at <https://api.ed-fi.org/v2.4.0/docs/>.

- In Windows Explorer, right-click on the downloaded ZIP file and select **Extract All...** Enter "C:\" for the target folder. The ZIP files contain an embedded folder ending in "-2.4.0" (i.e., C:\Ed-Fi-ODS-API-SDK-2.4.0).
- After the extractions are complete, rename the folder to remove the "-2.4.0" from the folder name (i.e., C:\Ed-Fi-ODS-API-SDK).

### Step 3. Build the Swagger Codegen JAR File

- Open a command prompt to C:\Ed-Fi-ODS-API-SDK and type "sbt assembly". This will build the Swagger Codegen JAR file in C:\Ed-Fi-ODS-API-SDK\target\scala-2.10 using the Scala Build Tool installed above.

### Step 4. Generate the SDK Source Files

The SDK source files are generated via a few simple command line steps.

```
java -jar sdk-generate.jar csharp --url https://{Domain root of API}
/metadata/{section}/api-docs --baseDir {destination folder} --apiPackage
{API namespace} --helperPackage {SDK namespace} --modelPackage {Model
namespace}
```

A brief description of the switch options follows:

- **target-language.** csharp or java (we'll use csharp).
- **--url.** https://{Domain root of API}/metadata/{section}/api-docs (where section is generally "resource", "descriptors", "types", or an API Profile or Composite Resource name).
- **--baseDir.** The local folder where you would like the code files to be created.
- **--apiPackage.** Namespace for the classes that expose methods that map to resources and verbs (e.g., EdFi.OdsApi.Api.{section}).
- **--modelPackage.** Namespace for the entities that are exchanged with the Ed-Fi ODS / API (e.g., EdFi.OdsApi.Models.{section}).
- **--helperPackage.** Namespace for the general SDK classes that support the Ed-Fi ODS / API interaction (e.g., EdFi.OdsApi.Sdk).
- **--projectName.** The name of the Visual Studio project for the generated SDK.

To generate SDK source files, navigate to C:\Ed-Fi-ODS-API-SDK\target\scala-2.10 and run the following commands to generate C# SDK source files for the Ed-Fi-hosted instance at <https://api.ed-fi.org>:

```
java -jar sdk-generate.jar csharp --url https://api.ed-fi.org/v2.4.0/api
/metadata/resources/api-docs --baseDir C:\Ed-Fi-ODS-API-SDK\CSharpSDK --
apiPackage EdFi.OdsApi.Api.Resources --modelPackage EdFi.OdsApi.Models.
Resources --helperPackage EdFi.OdsApi.Sdk --projectName EdFiClientSDK

java -jar sdk-generate.jar csharp --url https://api.ed-fi.org/v2.4.0/api
/metadata/descriptors/api-docs --baseDir C:\Ed-Fi-ODS-API-SDK\CSharpSDK --
apiPackage EdFi.OdsApi.Api.Descriptors --modelPackage EdFi.OdsApi.Models.
Descriptors --helperPackage EdFi.OdsApi.Sdk --projectName EdFiClientSDK

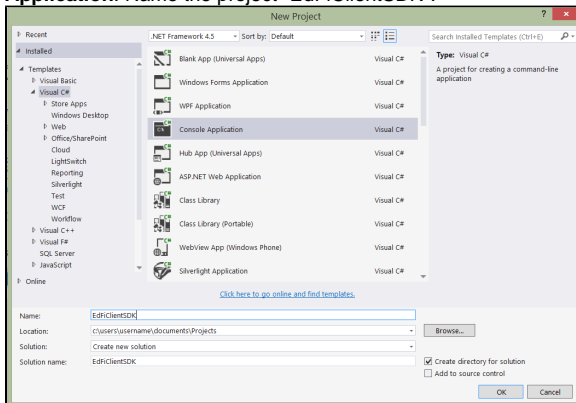
java -jar sdk-generate.jar csharp --url https://api.ed-fi.org/v2.4.0/api
/metadata/types/api-docs --baseDir C:\Ed-Fi-ODS-API-SDK\CSharpSDK --
apiPackage EdFi.OdsApi.Api.Types --modelPackage EdFi.OdsApi.Models.Types --
helperPackage EdFi.OdsApi.Sdk --projectName EdFiClientSDK

java -jar sdk-generate.jar csharp --url https://api.ed-fi.org/v2.4.0/api
/metadata/enrollment/api-docs --baseDir C:\Ed-Fi-ODS-API-SDK\CSharpSDK --
apiPackage EdFi.OdsApi.Api.EnrollmentComposite --modelPackage EdFi.OdsApi.
Models.EnrollmentComposite --helperPackage EdFi.OdsApi.Sdk --projectName
EdFiClientSDK

java -jar sdk-generate.jar csharp --url https://api.ed-fi.org/v2.4.0/api
/metadata/school-and-student/api-docs --baseDir C:\Ed-Fi-ODS-API-
SDK\CSharpSDK --apiPackage EdFi.OdsApi.Api.SchoolandStudent --modelPackage
EdFi.OdsApi.Models.School_and_Student --helperPackage EdFi.OdsApi.Sdk --
projectName EdFiClientSDK
```

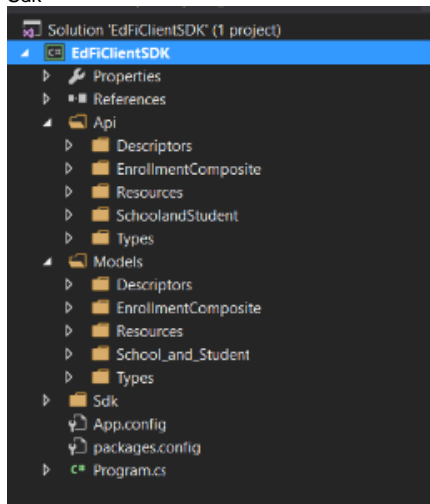
### Step 5. Use the SDK in a Sample C# Program

1. Open up Visual Studio and create a new project. Choose the type **Visual C# > Console Application**. Name the project "EdFiClientSDK".



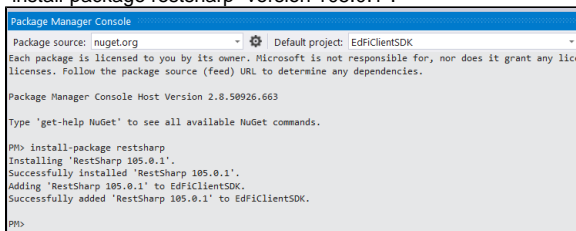
2. Add folders to the project, one for each of the folders generated from the previous step, i.e.,:

- Api
  - \Resources
  - \Descriptors
  - \Types
  - \EnrollmentComposite
  - \School\_and\_Student
- Models
  - \Resources
  - \Descriptors
  - \Types
  - \EnrollmentComposite
  - \School\_and\_Student
- Sdk



3. Right-click on each of the folders and add the existing classes for each package. The files can be found at the following location: C:\Ed-Fi-ODS-API-SDK\CSHarpSDK\EdFiClientSDK\EdFi\OdsApi. Be sure to select all of the generated SDK files for each package.

4. Use the Package Manager Console to install the RestSharp library. At the PM> prompt, enter "install-package restsharp -version 105.0.1".



5. Edit the Program.cs file and paste and add the following using statements at the top of the file:

```
using EdFi.OdsApi.Sdk;
using EdFi.OdsApi.Models.Resources;
using EdFi.OdsApi.Api.Resources;
using RestSharp;
```

6. Edit the Program.cs file and paste the following into the Main method. The client and key are using a publicly available sandbox environment with sample data hosted by the Ed-Fi Alliance.

```
// Trust all SSL certs -- needed unless signed SSL certificates are
configured.
System.Net.ServicePointManager.ServerCertificateValidationCallback =
    ((sender, certificate, chain, sslPolicyErrors) => true);

//Explicitly configures outgoing network calls to use the latest
version of TLS where possible.
//Due to our reliance on some older libraries, the.NET framework
won't necessarily default
//to the latest unless we explicitly request it. Some hosting
environments will not allow older versions
//of TLS, and thus calls can fail without this extra configuration.
System.Net.ServicePointManager.SecurityProtocol |= System.Net.
SecurityProtocolType.Tls11 | System.Net.SecurityProtocolType.Tls12;

// OAuth configuration
var oauthUrl = "https://api.ed-fi.org/v2.4.0/api/";
var clientKey = "RvcohKz9zHI4";
var clientSecret = "EliEFusaNf81xzCxwHfbolkC";

// RestSharp dependency, install via NuGet
var client = new RestClient("https://api.ed-fi.org/v2.4.0/api/api/v2.
0/2018/");

// TokenRetriever makes the OAuth calls
var tokenRetriever = new TokenRetriever(oauthUrl, clientKey,
clientSecret);

// Plug Oauth into RestSharp's authentication scheme
client.Authenticator = new BearerTokenAuthenticator(tokenRetriever);

// GET schools
var api = new SchoolsApi(client);

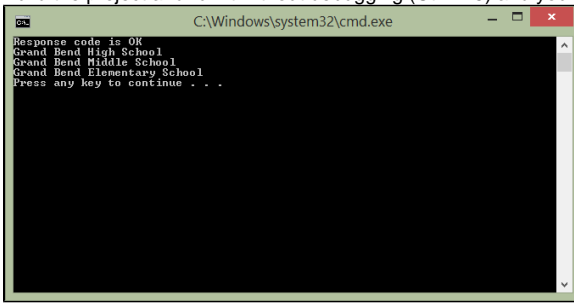
var response = api.GetSchoolsAll(null, null); // offset, limit

var httpReponseCode = response.StatusCode; // returns System.Net.
HttpStatusCode.OK
var schools = response.Data;

Console.WriteLine("Response code is " + httpReponseCode);

foreach (var school in schools)
{
    Console.WriteLine(school.nameOfInstitution);
}
Console.WriteLine();
Console.WriteLine("Hit ENTER key to continue...");
Console.ReadLine();
```

7. Build the project and run it without debugging (**Ctrl+F5**) and you should see the following results:



```
C:\Windows\system32\cmd.exe
Response code is OK
Grand Bend High School
Grand Bend Middle School
Grand Bend Elementary School
Press any key to continue . . .
```

With that, you're done!

This exercise leveraged a publicly available instance of the API, which contains the surface for a core implementation. If you're working with a specific platform host and you already have a key/secret pair, a great next step is to use these same techniques to generate an SDK for that platform. If the platform host has extended the data model, your new code will automatically include those structures in the data access components in the generated code.