

Using Code Generation to Create an SDK

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 Latest Version of Java](#)
- [Step 2. Download the OpenApi Codegen JAR File](#)
- [Step 3. Generate the SDK Source Files](#)
- [Step 4. Use the SDK in a Sample C# Program](#)

Each step is outlined in detail below.

Step 1. Install Latest Version of Java

If you don't already have Java installed, navigate to <https://java.com/en/download/> and download the latest installer. Run the installer to install the latest version of Java. In case you're wondering: the code generation leverages Java, but it does output C# code.

Step 2. Download the OpenApi Codegen JAR File

Download the latest version of the OpenApi Codegen JAR 6.0.1. Windows users can use Invoke-WebRequest in PowerShell 3.0+.

```
Invoke-WebRequest -OutFile openApi-codegen-cli.jar https://rep01.maven.org/maven2/org/openapitools/openapi-generator-cli/6.0.1/openapi-generator-cli-6.0.1.jar
```

For more information and download options visit <https://github.com/OpenAPITools/openapi-generator>.



When generating an SDK using the OpenApi CodeGen resources in a language other than C# or Java and there are profiles defined in the OpenAPI specification file, be sure to verify that the proper contentTypes were created during the code generation.

Step 3. Generate the SDK Source Files

The SDK source files are generated using Swagger metadata via a few simple PowerShell commands. You can see the available metadata endpoints for SDK generation at <https://api.ed-fi.org/v6.0/api/metadata?sdk=true>.

```
java -jar <openapi-codegen-jar-file> generate -g csharp-netcore -i <target-swagger-json-file> --additional-properties targetFramework=net6.0,netCoreProjectFile=true --skip-validate-spec
```

A detailed description of the switch options can be found at <https://github.com/OpenAPITools/openapi-generator>.

To generate SDK source files, navigate to the folder containing openapi-codegen-cli.jar and run the following commands in PowerShell 3.0+ to generate C# SDK source files. You can use the unified SDK generation endpoints to combine resources, descriptors, and extensions under one namespace/directory in the generated SDK.

Resources, Descriptors and Extensions

```
java -jar openApi-codegen-cli.jar generate -g csharp-netcore -i https://api.ed-fi.org/v6.0/api/metadata/data/v3/swagger.json --api-package Apis.All --model-package Models.All -o ./csharp --additional-properties packageName=EdFi.OdsApi.Sdk,targetFramework=net6.0,netCoreProjectFile=true --global-property modelTests=false --global-property apiTests=false --skip-validate-spec
```

Downloads

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

[Ed-Fi-ODS-API-SDK.zip](#)

The Sample program works against the Ed-Fi ODS / API sandbox hosted at <https://api.ed-fi.org/v6.0/docs/>.

Ed-Fi Core Resources and Descriptors

```
java -jar openApi-codegen-cli.jar generate -g csharp-netcore -i
https://api.ed-fi.org/v6.0/api/metadata/data/v3/ed-fi/swagger.json --api-
package Apis.All --model-package Models.All -o ./csharp --additional-
properties packageName=EdFi.OdsApi.Sdk,targetFramework=net6.0,
netCoreProjectFile=true --global-property modelTests=false --global-
property apiTests=false --skip-validate-spec
```

Alternatively, you can generate SDK with segregated namespaces for resources, descriptors, enrollment composites, and Identity API endpoints as follows:

```
java -jar openApi-codegen-cli.jar generate -g csharp-netcore -i
https://api.ed-fi.org/v6.0/api/metadata/data/v3/resources/swagger.json --
api-package Api.Resources --model-package Models.Resources -o ./csharp --
additional-properties packageName=EdFi.OdsApi.Sdk,targetFramework=net6.0,
netCoreProjectFile=true --global-property modelTests=false --global-
property apiTests=false --skip-validate-spec

java -jar openApi-codegen-cli.jar generate -g csharp-netcore -i
https://api.ed-fi.org/v6.0/api/metadata/composites/v1/ed-fi/enrollment
/swagger.json --api-package Api.EnrollmentComposites --model-package
Models.EnrollmentComposites -o ./csharp --additional-properties
packageName=EdFi.OdsApi.Sdk,targetFramework=net6.0,netCoreProjectFile=true
--global-property modelTests=false --global-property apiTests=false --skip-
validate-spec

java -jar openApi-codegen-cli.jar generate -g csharp-netcore -i
https://api.ed-fi.org/v6.0/api/metadata/identity/v2/swagger.json --api-
package Api.Identities --model-package Models.Identities -o ./csharp --
additional-properties packageName=EdFi.OdsApi.Sdk,targetFramework=net6.0,
netCoreProjectFile=true --global-property modelTests=false --global-
property apiTests=false --skip-validate-spec

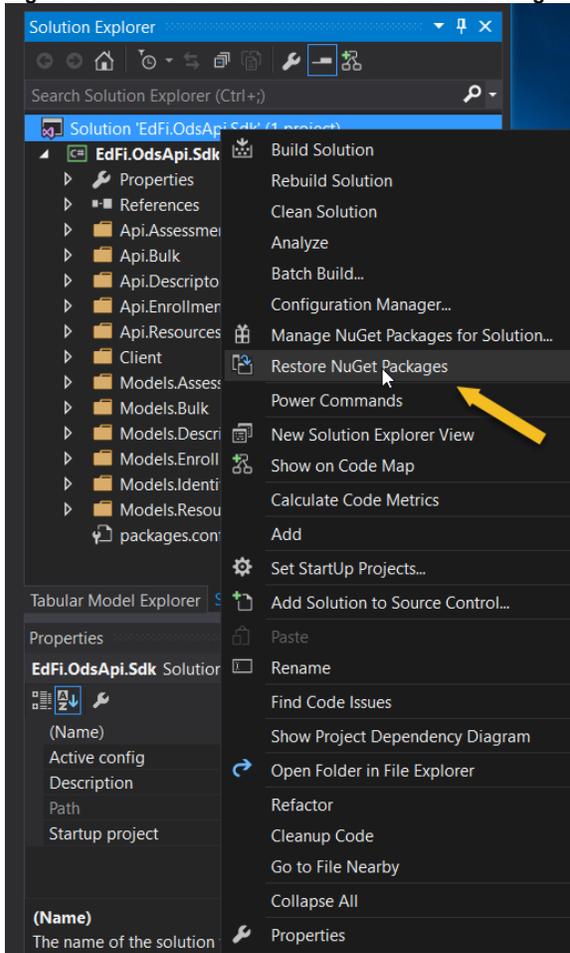
java -jar openApi-codegen-cli.jar generate -g csharp-netcore -i
https://api.ed-fi.org/v6.0/api/metadata/data/v3/descriptors/swagger.json --
api-package Api.Descriptors --model-package Models.Descriptors -o ./csharp
--additional-properties packageName=EdFi.OdsApi.Sdk,targetFramework=net6.0,
netCoreProjectFile=true --global-property modelTests=false --global-
property apiTests=false --skip-validate-spec
```

Wait for the Swagger CodeGen to finish generating code. A Visual Studio Solution named **EdFi.OdsApi.Sdk.sln** will be created with the SDK artifacts.

Step 4. Use the SDK in a Sample C# Program

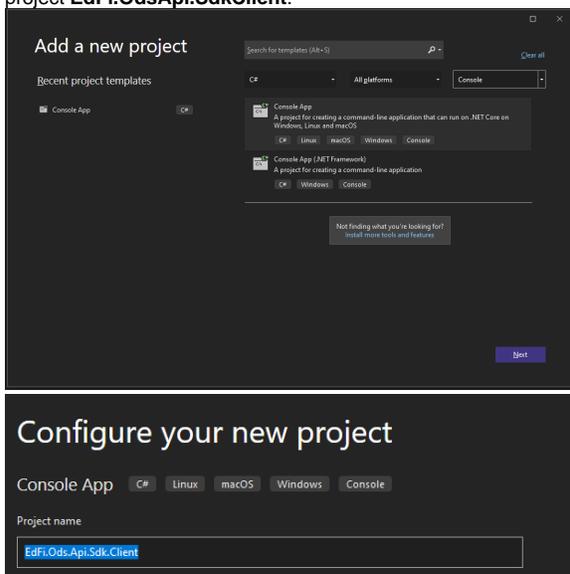
1. **Open** EdFi.OdsApi.Sdk.sln in Visual Studio.

2. **Right-click** on the solution and **click Restore NuGetPackages.**

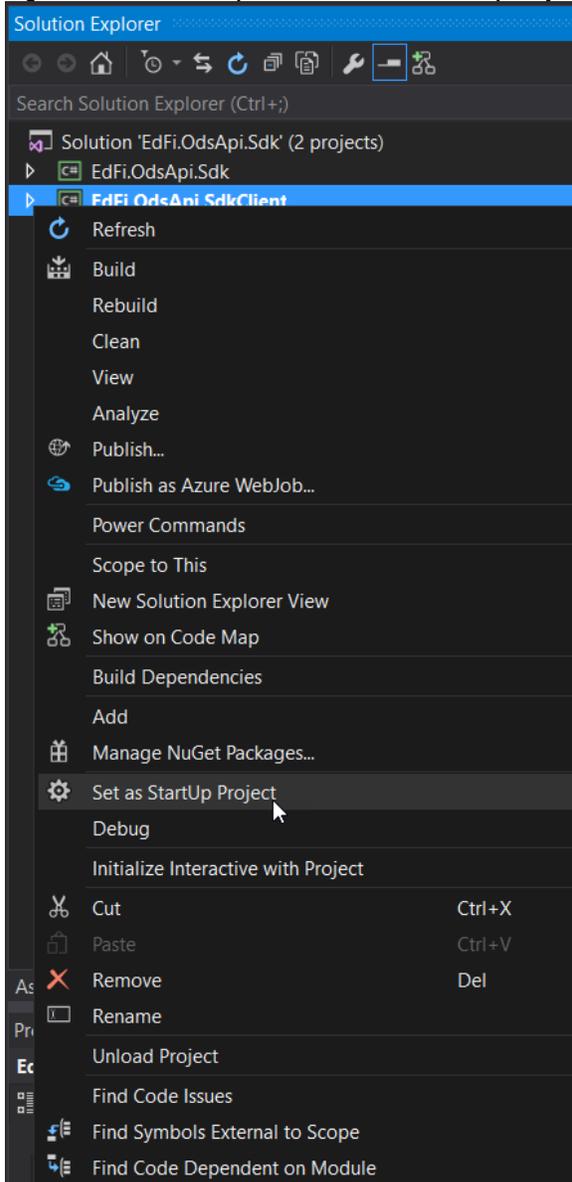


3. **Build** the solution.

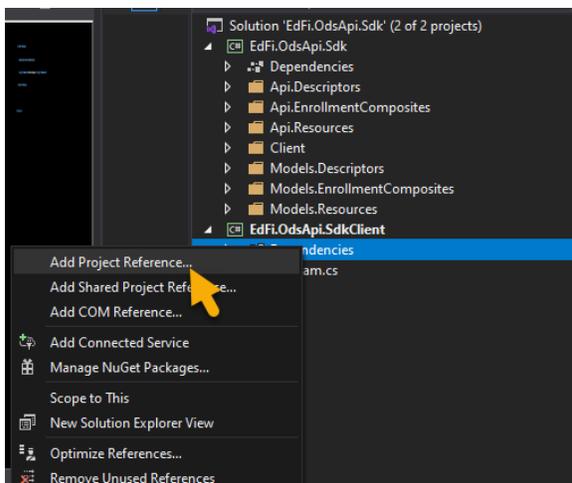
4. **Right-click** on the solution and **add a new project.** Choose the type **Console App.** Name the project **EdFi.OdsApi.SdkClient.**



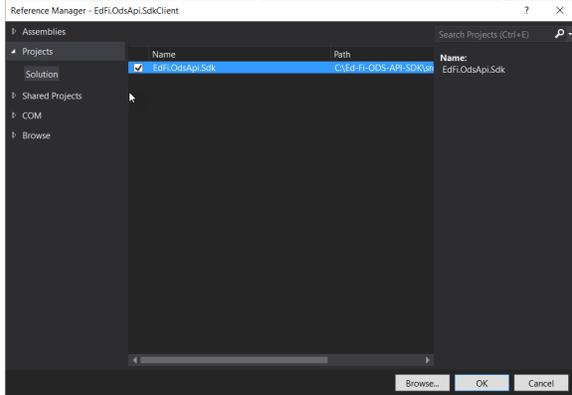
5. Right-click on Edfi.OdsApi.SdkClient > Set as Startup Project.



6. In Solution Explorer, right-click EdFi.OdsApi.SdkClient Dependencies node and click Add Project Reference.



- In the **Add Reference > Projects** tab select **EdFi.OdsApi.Sdk**, and then click **OK**.



- In **Solution Explorer**, right-click **EdFi.OdsApi.SdkClient Dependencies** node and click **Manage NuGet Packages...** and add RestSharp reference.
- Open the **Program.cs** file and add the following **using** statements at the top of the file:

```
using EdFi.OdsApi.Sdk.Apis.All;  
using EdFi.OdsApi.Sdk.Client;  
using EdFi.OdsApi.Sdk.Models.All;  
using EdFi.OdsApi.SdkClient;
```

- Edit the **Program.cs** file and paste the following code. 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/v6.0/api/";  
var clientKey = "RvcohKz9zHI4";  
var clientSecret = "EliEFusaNf81xzCxwHfbolkC";  
  
// TokenRetriever makes the oauth calls. It has RestSharp  
// dependency, install via NuGet  
var tokenRetriever = new TokenRetriever(oauthUrl, clientKey,  
    clientSecret);  
  
// Plug Oauth access token. Tokens will need to be refreshed when  
// they expire  
var configuration = new Configuration() { AccessToken =  
    tokenRetriever.ObtainNewBearerToken(), BasePath = "https://api.ed-fi.  
org/v6.0/api/data/v3" };  
  
// GET students  
var apiInstance = new StudentsApi(configuration);  
apiInstance.Configuration.DefaultHeaders.Add("Content-Type",  
    "application/json");  
  
// Fetch a single record with the totalCount flag set to true to  
// retrieve the total number of records available  
var studentWithHttpInfo = apiInstance.GetStudentsWithHttpInfo(limit:  
    1, offset: 0, totalCount: true);
```

```

var httpReponseCode = studentWithHttpInfo.StatusCode; // returns
System.Net.HttpStatusCode.OK
Console.WriteLine("Response code is " + httpReponseCode);

// Parse the total count value out of the "Total-Count" response
header
int.TryParse(studentWithHttpInfo.Headers["Total-Count"].First(), out
var totalCount);

int offset = 0;
int limit = 100;
var students = new List<EdFiStudent>();

while (offset < totalCount)
{
    Console.WriteLine($"Fetching student records {offset}
through {Math.Min(offset + limit, totalCount)} of {totalCount}");
    students.AddRange(apiInstance.GetStudents(limit: limit,
offset: 0));
    offset += limit;
}

Console.WriteLine();
Console.WriteLine("Student Results");

foreach (var student in students)
{
    Console.WriteLine($"Student: {student.StudentUniqueId},
{student.FirstName}, {student.LastSurname}");
}

Console.WriteLine();
Console.WriteLine("Hit ENTER key to continue...");
Console.ReadLine();

```

11. Add a **.cs** file named **TokenRetriever.cs** and copy the following code to help with OAuth integration.

```

using System.Net;
using System.Runtime.Serialization;
using System.Security.Authentication;
using EdFi.OdsApi.Sdk.Client;

namespace EdFi.OdsApi.SdkClient
{
    public class TokenRetriever
    {
        private string oauthUrl;
        private string clientKey;
        private string clientSecret;

        public TokenRetriever(string oauthUrl, string clientKey,
string clientSecret)
        {
            this.oauthUrl = oauthUrl;
            this.clientKey = clientKey;
            this.clientSecret = clientSecret;
        }

        public string ObtainNewBearerToken()
        {
            var oauthClient = new ApiClient(oauthUrl);
            return GetBearerToken(oauthClient);
        }

        private string GetBearerToken(ApiClient oauthClient)
        {
            var configuration = new Configuration() { BasePath =
oauthUrl };

```

```

        var bearerTokenRequestOptions = new RequestOptions() {
Operation = String.Empty };
        bearerTokenRequestOptions.FormParameters.Add
("Client_id", clientKey);
        bearerTokenRequestOptions.FormParameters.Add
("Client_secret", clientSecret);
        bearerTokenRequestOptions.FormParameters.Add
("Grant_type", "client_credentials");

        var bearerTokenResponse = oauthClient.
Post<BearerTokenResponse>("oauth/token", bearerTokenRequestOptions,
configuration);
        if (bearerTokenResponse.StatusCode != HttpStatusCode.OK)
        {
            throw new AuthenticationException("Unable to
retrieve an access token. Error message: " +
bearerTokenResponse.Data.Error);
        }

        if (bearerTokenResponse.Data.Error != null ||
bearerTokenResponse.Data.TokenType != "bearer")
        {
            throw new AuthenticationException(
                "Unable to retrieve an access token. Please
verify that your application secret is correct.");
        }

        return bearerTokenResponse.Data.AccessToken;
    }
}

[DataContract]
internal class BearerTokenResponse
{
    [DataMember(Name = "access_token", EmitDefaultValue = false)]
    public string AccessToken { get; set; }

    [DataMember(Name = "expires_in", EmitDefaultValue = false)]
    public string ExpiresIn { get; set; }

    [DataMember(Name = "token_type", EmitDefaultValue = false)]
    public string TokenType { get; set; }

    [DataMember(Name = "error", EmitDefaultValue = false)]
    public string Error { get; set; }
}
}

```

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

```

Fetching student records 0 through 100 of 960
Fetching student records 100 through 200 of 960
Fetching student records 200 through 300 of 960
Fetching student records 300 through 400 of 960
Fetching student records 400 through 500 of 960
Fetching student records 500 through 600 of 960
Fetching student records 600 through 700 of 960
Fetching student records 700 through 800 of 960
Fetching student records 800 through 900 of 960
Fetching student records 900 through 960 of 960

Student Results
Student: 604821, Tyrone, Dyer
Student: 604823, Julie, Randolph
Student: 604824, Traci, Mathews
Student: 604822, Lisa, Woods
Student: 604826, Jeremy, Coleman
Student: 604825, Dale, Jimenez
Student: 604827, Vincent, Orozco
Student: 604828, Julie, Beard
Student: 604831, Peter, Pratt
Student: 604829, April, Shelton
Student: 604830, Rick, Owen
Student: 604832, Beverly, Ortiz
Student: 604833, Dennis, Mac Donald
Student: 604834, James, Mathews

```

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, 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.