

How To: Use Swagger Codegen Tools to Create a Client 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 Swagger 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 Swagger Codegen JAR File

Download the latest version of the Swagger Codegen JAR 2.3.0+. Windows users can use Invoke-WebRequest in PowerShell 3.0+.

```
Invoke-WebRequest -OutFile swagger-codegen-cli.jar http://central.maven.org/maven2/io/swagger/swagger-codegen-cli/2.3.0/swagger-codegen-cli-2.3.0.jar
```

For more information and download options visit <https://github.com/swagger-api/swagger-codegen>.

Step 3. Generate the SDK Source Files

The SDK source files are generated via a few simple PowerShell commands.

```
java -jar <swagger-codegen-jar-file> generate -l csharp -i <target-swagger-json-file>
```

A detailed description of the switch options can be found at <https://github.com/swagger-api/swagger-codegen>.

To generate SDK source files, navigate to the folder containing swagger-codegen-cli.jar and run the following commands in PowerShell 3.0+ to generate C# SDK source files for the Ed-Fi-hosted instance at <https://api.ed-fi.org/v3/api/>.

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.

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/v3/docs/>.

```

java -jar swagger-codegen-cli.jar generate -l csharp -i https://api.ed-fi.org/v3/api/metadata/data/v3/resources/swagger.json --api-package Api.Resources --model-package Models.Resources -DmodelTests=false -DapiTests=false -DpackageName='EdFi.OdsApi.Sdk'

java -jar swagger-codegen-cli.jar generate -l csharp -i https://api.ed-fi.org/v3/api/metadata/composites/v1/ed-fi/enrollment/swagger.json --api-package Api.EnrollmentComposites --model-package Models.EnrollmentComposites -DmodelTests=false -DapiTests=false -DpackageName='EdFi.OdsApi.Sdk'

java -jar swagger-codegen-cli.jar generate -l csharp -i https://api.ed-fi.org/v3/api/metadata/bulk/v1/swagger.json --api-package Api.Bulk --model-package Models.Bulk -DmodelTests=false -DapiTests=false -DpackageName='EdFi.OdsApi.Sdk'

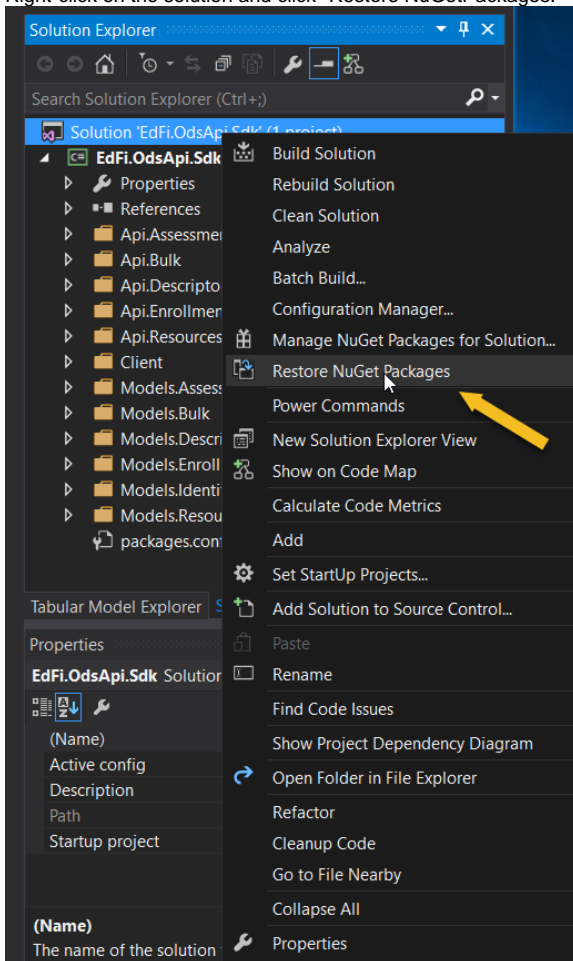
java -jar swagger-codegen-cli.jar generate -l csharp -i https://api.ed-fi.org/v3/api/metadata/identity/v2/swagger.json --api-package Api.Identities --model-package Models.Identities -DmodelTests=false -DapiTests=false -DpackageName='EdFi.OdsApi.Sdk'

java -jar swagger-codegen-cli.jar generate -l csharp -i https://api.ed-fi.org/v3/api/metadata/data/v3/descriptors/swagger.json --api-package Api.Descriptors --model-package Models.Descriptors -DmodelTests=false -DapiTests=false -DpackageName='EdFi.OdsApi.Sdk'

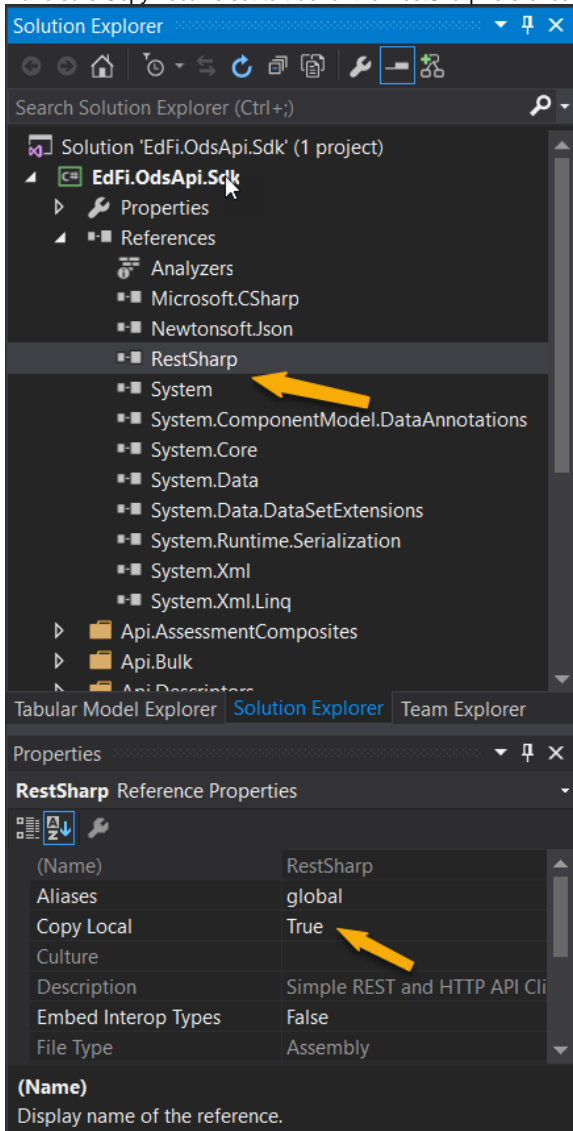
```

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

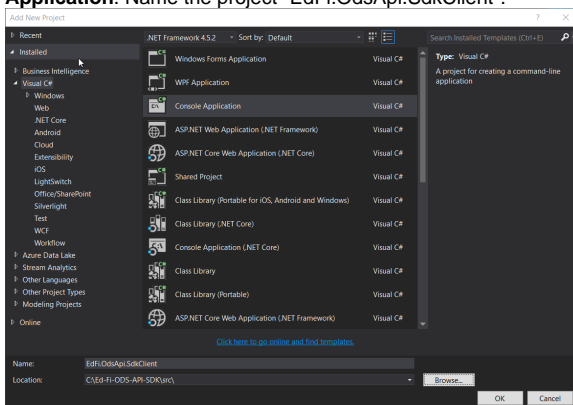
1. Open EdFi.OdsApi.Sdk.sln in Visual Studio.
2. Right-click on the solution and click "Restore NuGet Packages."



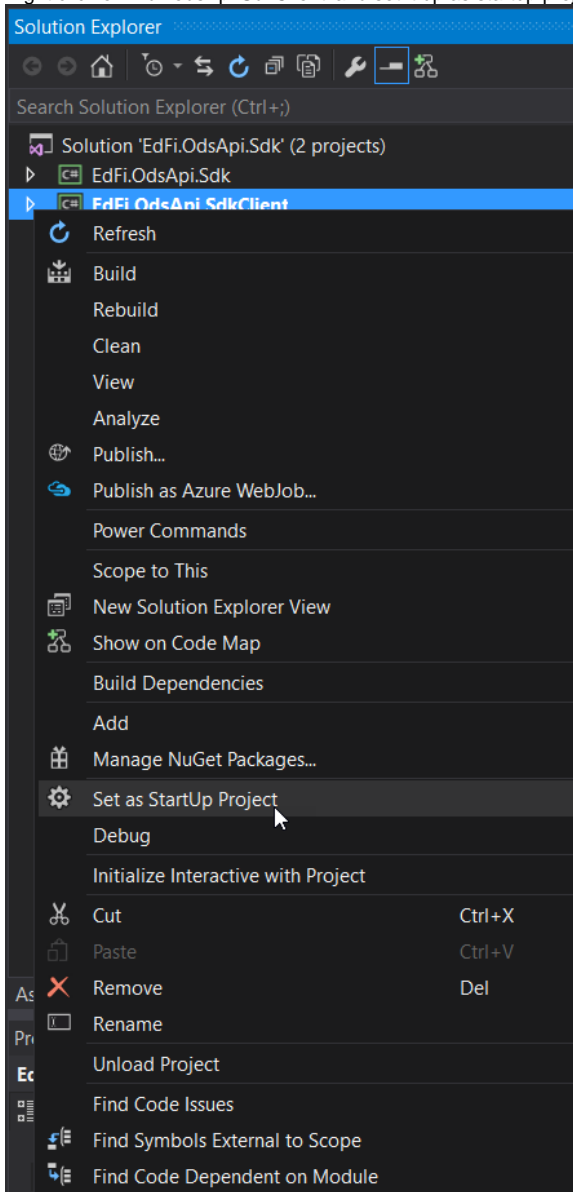
3. Make sure Copy Local is set to true for the RestSharp reference.



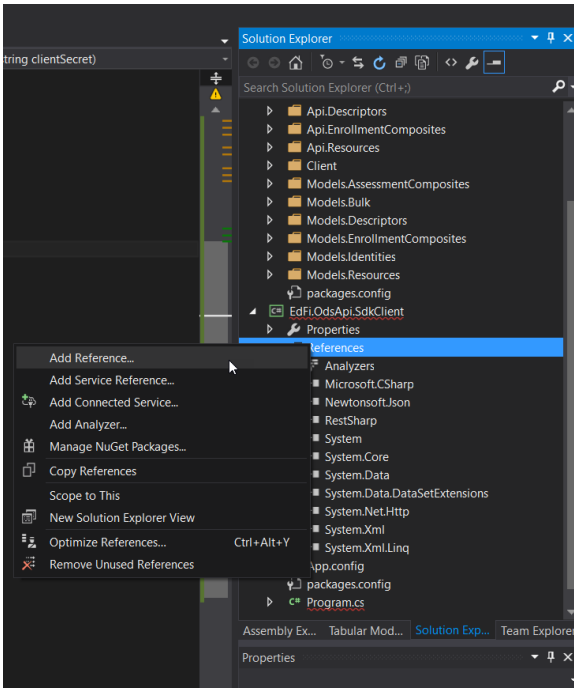
4. Build the solution.
5. Right-click on the solution and add a new project. Choose the type **Visual C# > Console Application**. Name the project "EdFi.OdsApi.SdkClient".



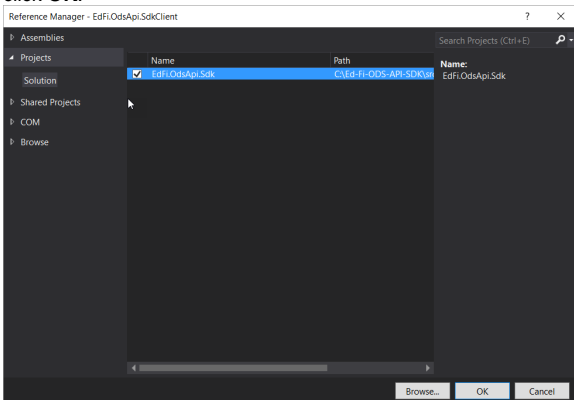
6. Right click on 'Edfi.odsApi.SdkClient' and set it up as startup project.



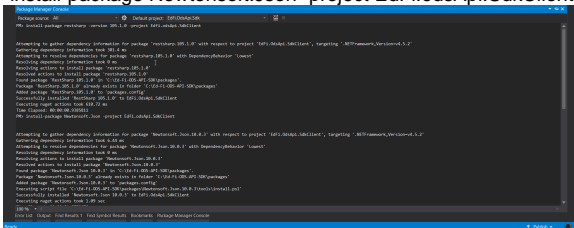
7. In **Solution Explorer**, right-click on the "EdFi.OdsApi.SdkClient" project references node and click **Add Reference**.



- In the **Add Reference** dialog box, select the projects tab and select EdFi.OdsApi.Sdk, and then click **OK**.



- Use the Package Manager Console to install the RestSharp and Newtonsoft libraries. At the **PM>** prompt, enter "install-package restsharp -version 105.1.0 -project EdFi.odsApi.SdkClient" and "install-package Newtonsoft.Json -project EdFi.odsApi.SdkClient"



- Open the Program.cs file and add the following `using` statements at the top of the file:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using EdFi.OdsApi.Sdk.Api.Resources;
using EdFi.OdsApi.Sdk.Client;
```

- 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/v3/api";
var clientKey = "RvcohKz9zHI4";
var clientSecret = "EliEFusaNf8lxzCxwHfbolkC";

// 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:443/v3/api/data/v3" };

// GET schools
var api = new SchoolsApi(configuration);
var response = api.GetSchoolsWithHttpInfo(null, null); // offset,
limit
var httpResponseCode = response.StatusCode; // returns System.Net.
HttpStatusCode.OK
var schools = response.Data;

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

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

```

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

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Net;
using System.Net.Http;
using System.Net.Http.Headers;
using System.Security.Authentication;
using System.Text;
using System.Threading.Tasks;
using Newtonsoft.Json;
using RestSharp;

namespace EdFi.OdsApi.SdkClient
{
    public class TokenRetriever
    {

```

```

private string oauthUrl;
private string clientKey;
private string clientSecret;

/// <summary>
///
/// </summary>
/// <param name="oauthUrl"></param>
/// <param name="clientKey"></param>
/// <param name="clientSecret"></param>
public TokenRetriever(string oauthUrl, string clientKey,
string clientSecret)
{
    this.oauthUrl = oauthUrl;
    this.clientKey = clientKey;
    this.clientSecret = clientSecret;
}

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

private string GetBearerToken(IRestClient oauthClient)
{
    var bearerTokenRequest = new RestRequest("oauth/token",
Method.POST);
    bearerTokenRequest.AddParameter("Client_id", clientKey);
    bearerTokenRequest.AddParameter("Client_secret",
clientSecret);
    bearerTokenRequest.AddParameter("Grant_type",
"client_credentials");

    var bearerTokenResponse = oauthClient.
Execute<BearerTokenResponse>(bearerTokenRequest);
    if (bearerTokenResponse.StatusCode != HttpStatusCode.OK)
    {
        throw new AuthenticationException("Unable to
retrieve an access token. Error message: " +
bearerTokenResponse.ErrorMessage);
    }

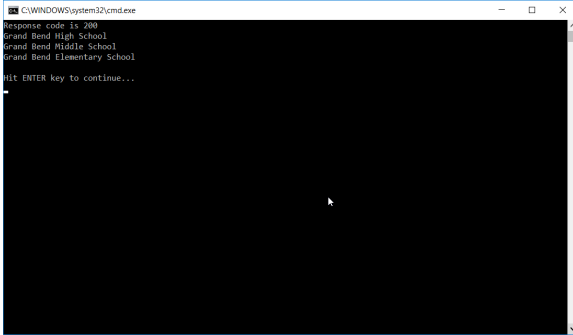
    if (bearerTokenResponse.Data.Error != null ||
bearerTokenResponse.Data.TokenType != "bearer")
    {
        throw new AuthenticationException(
            "Unable to retrieve an access token. Please
verify that your application secret is correct.");
    }

    return bearerTokenResponse.Data.AccessToken;
}
}

internal class BearerTokenResponse
{
    public string AccessToken { get; set; }
    public string ExpiresIn { get; set; }
    public string TokenType { get; set; }
    public string Error { get; set; }
}
}

```

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



```
COMNDOWS\system32\cmd.exe
Response code is 200
Grand Bend High School
Grand Bend Middle School
Grand Bend Elementary School
Hit ENTER 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, 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.