What is Swagger?

[Swagger](https://swagger.io/swagger-ui/) (now known as the [OpenAPI](https://www.openapis.org/)) is a framework for describing your API by using a common language that is easy read and understand by developers and testers.

## What is Swagger UI?

[Swagger UI](https://swagger.io/swagger-ui/), a part of Swagger, is an open source tool that generates a web page that documents the APIs generated by the Swagger specification. This UI presentation of the APIs is user friendly and easy to understand. This enables developers to execute and monitor the API requests they sent and the results they received. Swagger UI represents apis within browser so brings more intuitively than other tools such as: [Postman](https://www.blazemeter.com/blog/getting-started-postman?utm_source=blog&utm_medium=BM_blog&utm_campaign=getting-started-with-swagger-ui), SoapUI, and [others](https://www.blazemeter.com/api-functional-testing?utm_source=blog&utm_medium=BM_blog&utm_campaign=getting-started-with-swagger-ui).

[NSwag](http://nswag.org/) (Swagger toolchain for .NET) so that you can access your Web API actions with the [Swagger UI](http://swagger.io/swagger-ui/) and a generated C# client library.

The toolchain can be integrated into the following three types of Web API projects:

* **Global.asax**-based Web projects (full .NET Framework)
* **OWIN**-based Web projects (full .NET Framework)
* **ASP.NET Core** based projects (full .NET Framework or .NET Core)

For each project, the integration is a little bit different, but this tutorial will provide links to all supported ways. Just follow the steps.

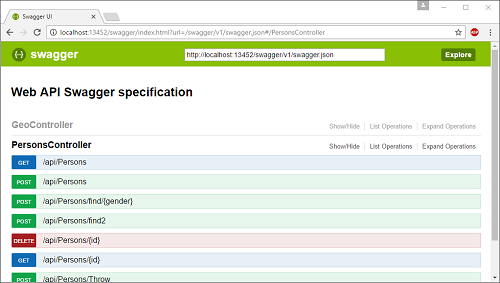
**1. Create and setup a new Web API project**

First we implement your Web API project:

1. Open Visual Studio and create a new Web API project (either Global.asax-, OWIN- or ASP.NET Core-based).
2. Register the NSwag Swagger middlewares:
   * **Global.asax:** [Register the OWIN Middlewares in your Global.asax](https://github.com/NSwag/NSwag/wiki/OwinGlobalAsax#integration).
   * **OWIN and ASP.NET Core:** [Register the OWIN or ASP.NET Core Middlewares in your Startup.cs](https://github.com/NSwag/NSwag/wiki/Middlewares#integration).
3. (optional) [Globally register the Web API exception serialization filter](https://github.com/NSwag/NSwag/wiki/JsonExceptionFilterAttribute#register-globally)

**2. Run the Web API project**

Run the project and navigate to the URL http://yourserver/swagger: Here you should see the Swagger UI.



Try calling your Web API actions and check the results.

**3. Generate client code to access the Web API**

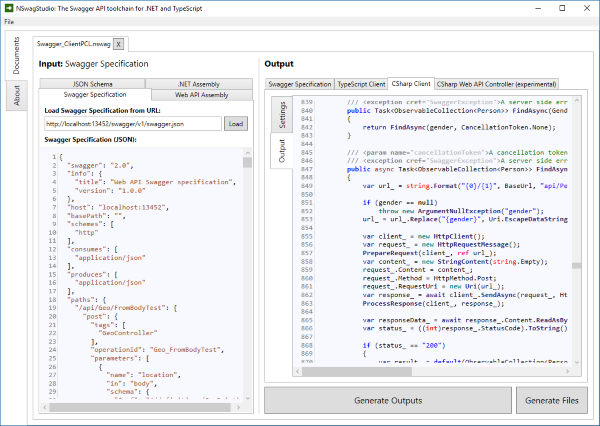
The next steps show how to generate a client library to access the Web API actions in the Web project. In this tutorial we use the C# client generator (there is also a TypeScript client generator).

**Setup tooling and create client project:**

1. Install [NSwagStudio](https://github.com/NSwag/NSwag/wiki/NSwagStudio).
2. Create a new C# library project in your solution (the service client library).
3. [Add the required assembly dependencies to the library project](https://github.com/NSwag/NSwag/wiki/SwaggerToCSharpClientGenerator#required-dependencies-for-the-client).

**Generate code:**

1. Start NSwagStudio and select “Swagger Specification” as input.
2. Enter the Swagger specification URL (default: http://yourserver/swagger/v1/swagger.json, the server must be running).
3. Click “Generate Outputs” and select the tab “CSharpClient”.
4. Copy the source code into your client project.
5. Now you should change generator settings in the NSwagStudio to improve and customize the generated code (e.g. define the correct namespace, etc.).



**Automate script generation:**

1. Save the .nswag file in NSwagStudio
2. Open a command prompt and cd to the directory with the .nswag file
3. Run nswag run to execute all .nswag files (the NSwag command line tool is automatically registered by the MSI installer)

**4. Have fun**

Now that you have seen a way to integrate NSwag into your project, start having fun with NSwag!

But there is always more: [Other ways to integrate and automate NSwag](https://github.com/NSwag/NSwag/wiki#ways-to-use-the-toolchain)