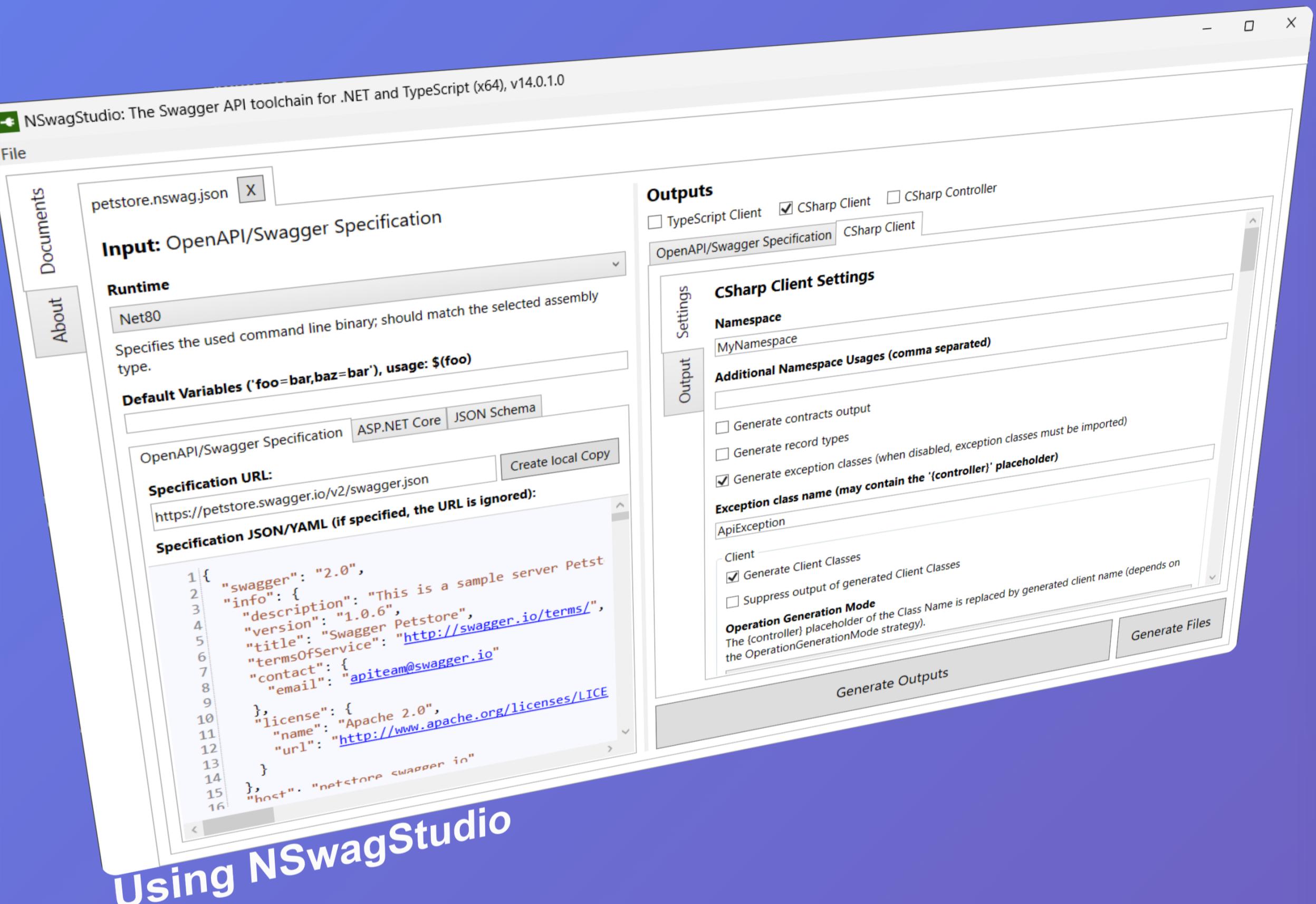


Generate code with NSwag using the OpenAPI Specification



Using NSwagStudio

```
# 🌟 created with NSwagStudio
nswag run ./petstore.nswag.json /runtime:Net70
```

Using the CLI

```
await GenerateCSharpClient(
    "https://petstore.swagger.io/v2/swagger.json",
    "PetStoreClient"
);

static async Task GenerateCSharpClient(string openApiLocation, string className)
{
    var document = await OpenApiDocument.FromUrlAsync(openApiLocation);
    var settings = new CSharpClientGeneratorSettings
    {
        UseBaseUrl = false,
        ClassName = className,
        GenerateClientInterfaces = true,
        CSharpGeneratorSettings =
        {
            Namespace = "HttpClients",
        },
    };

    var generator = new CSharpClientGenerator(document, settings);
    var generatedCode = generator.GenerateFile();

    var path = $"{settings.CSharpGeneratorSettings.Namespace}/{settings.ClassName}";
    var file = new FileInfo(path);
    file.Directory.Create();
    await File.WriteAllTextAsync(file.FullName, generatedCode);
}
```

Using code (NSwag.CodeGeneration.CSharp)



timdeschryver.dev/bits

Generate code with NSwag using the OpenAPI Specification

NSwagStudio: The Swagger API toolchain for .NET and TypeScript (x64), v14.0.1.0

File

Documents

About

Input: OpenAPI/Swagger Specification

Runtime
Net80

Specifies the used command line binary; should match the selected assembly type.

Default Variables ('foo=bar,baz=bar'), usage: \$(foo)

OpenAPI/Swagger Specification ASP.NET Core JSON Schema

Specification URL:
<https://petstore.swagger.io/v2/swagger.json> Create local Copy

Specification JSON/YAML (if specified, the URL is ignored):

```
1: {  
2:   "swagger": "2.0",  
3:   "info": {  
4:     "description": "This is a sample server Petst",  
5:     "version": "1.0.6",  
6:     "title": "Swagger Petstore",  
7:     "termsOfService": "http://swagger.io/terms/",  
8:     "contact": {  
9:       "email": "apiteam@swagger.io"  
10:    },  
11:    "license": {  
12:      "name": "Apache 2.0",  
13:      "url": "http://www.apache.org/licenses/LICENSE  
14:    }  
15:  },  
16:  "host": "petstore.swagger.io"
```

Outputs

TypeScript Client CSharp Client CSharp Controller

OpenAPI/Swagger Specification CSharp Client

CSharp Client Settings

Settings

Output

Namespace
MyNamespace

Additional Namespace Usages (comma separated)

Generate contracts output
 Generate record types
 Generate exception classes (when disabled, exception classes must be imported)

Exception class name (may contain the '{controller}' placeholder)
ApiException

Client

Generate Client Classes
 Suppress output of generated Client Classes

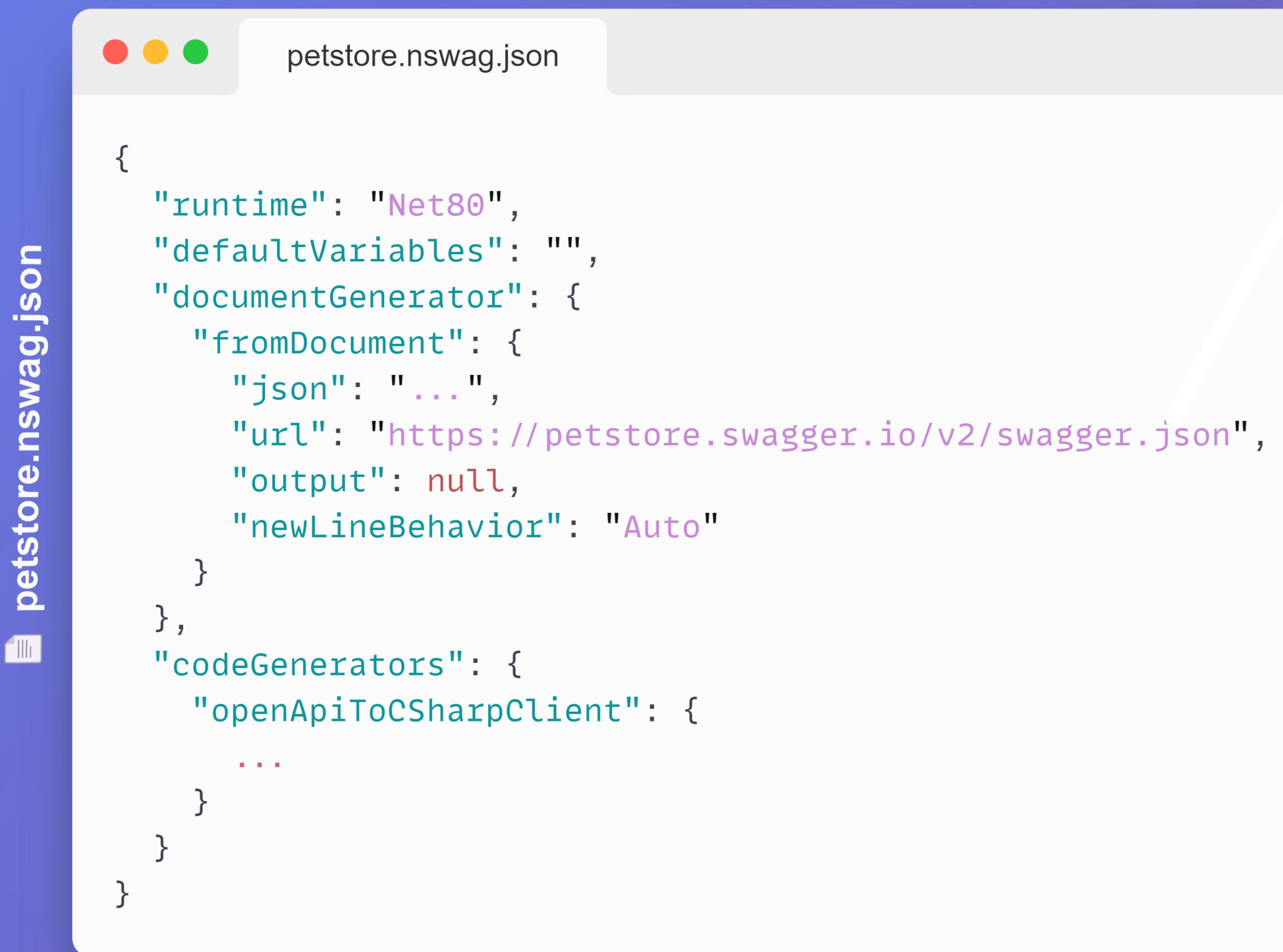
Operation Generation Mode
The {controller} placeholder of the Class Name is replaced by generated client name (depends on the OperationGenerationMode strategy).

Generate Outputs Generate Files

 petstore.nswag.json



Generate code with NSwag using the OpenAPI Specification



```
petstore.nswag.json
```

```
{  
  "runtime": "Net80",  
  "defaultVariables": "",  
  "documentGenerator": {  
    "fromDocument": {  
      "json": "...",  
      "url": "https://petstore.swagger.io/v2/swagger.json",  
      "output": null,  
      "newLineBehavior": "Auto"  
    }  
  },  
  "codeGenerators": {  
    "openApiToCSharpClient": {  
      ...  
    }  
  }  
}
```



```
# 🙌 created with NSwagStudio  
nswag run ./petstore.nswag.json /runtime:Net70
```

Generate code with NSwag using the OpenAPI Specification

```
await GenerateCSharpClient("https://petstore.swagger.io/v2/swagger.json", "PetStoreClient");

static async Task GenerateCSharpClient(string openApiLocation, string className)
{
    var document = await OpenApiDocument.FromUrlAsync(openApiLocation);
    var settings = new CSharpClientGeneratorSettings
    {
        UseBaseUrl = false,
        ClassName = className,
        GenerateClientInterfaces = true,
        CSharpGeneratorSettings =
        {
            Namespace = "HttpClients",
        },
    };

    var generator = new CSharpClientGenerator(document, settings);
    var generatedCode = generator.GenerateFile();

    var path = $"{settings.CSharpGeneratorSettings.Namespace}/{settings.ClassName}";
    var file = new FileInfo(path);
    file.Directory.Create();
    await File.WriteAllTextAsync(file.FullName, generatedCode);
}
```

Generate code with NSwag using the OpenAPI Specification

The generated code uses the OpenAPI Specification to create DTO models, HTTP Clients, and includes summaries.



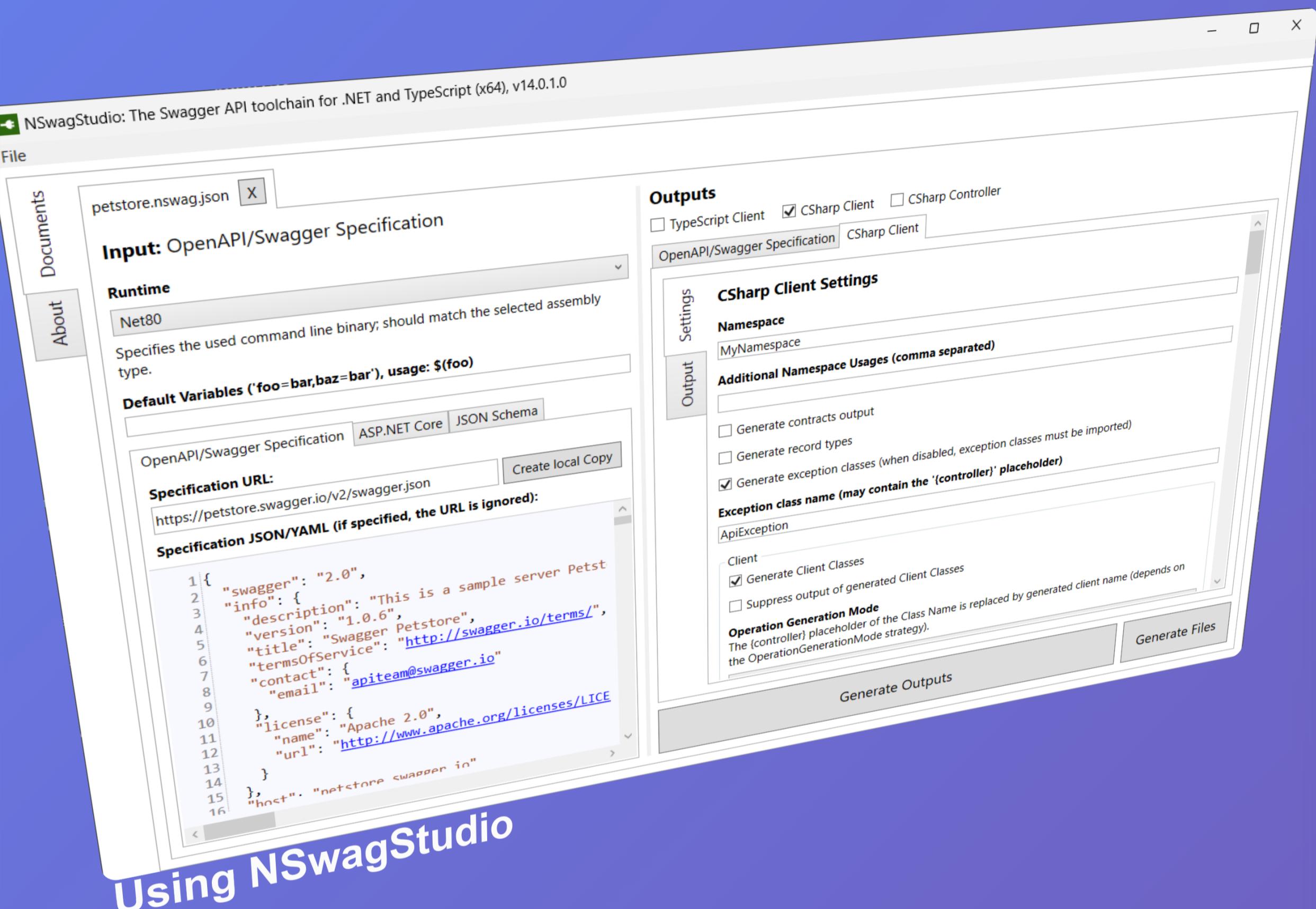
```
[System.CodeDom.Compiler.GeneratedCode("NSwag", "14.0.2.0 (NJsonSchema v11.0.0.0 (Newtonsoft.Json v13.0.0.0))")]
public partial interface IPetStoreClient
{
    /// <param name="cancellationToken">A cancellation token that can be used by other objects or threads to receive notice of cancellation.</param>
    /// <summary>
    /// Add a new pet to the store
    /// </summary>
    /// <param name="body">Pet object that needs to be added to the store</param>
    /// <exception cref="ApiException">A server side error occurred.</exception>
    System.Threading.Tasks.Task AddPetAsync(Pet body, System.Threading.CancellationToken cancellationToken);

    /// <param name="cancellationToken">A cancellation token that can be used by other objects or threads to receive notice of cancellation.</param>
    /// <summary>
    /// Find pet by ID
    /// </summary>
    /// <remarks>
    /// Returns a single pet
    /// </remarks>
    /// <param name="petId">ID of pet to return</param>
    /// <returns>successful operation</returns>
    /// <exception cref="ApiException">A server side error occurred.</exception>
    System.Threading.Tasks.Task<Pet> GetPetByIdAsync(long petId, System.Threading.CancellationToken cancellationToken);

    ...
}
```

Generated code

Generate code with NSwag using the OpenAPI Specification



Using NSwagStudio

```
# 🌟 created with NSwagStudio
nswag run ./petstore.nswag.json /runtime:Net70
```

Using the CLI

```
await GenerateCSharpClient(
    "https://petstore.swagger.io/v2/swagger.json",
    "PetStoreClient"
);

static async Task GenerateCSharpClient(string openApiLocation, string className)
{
    var document = await OpenApiDocument.FromUrlAsync(openApiLocation);
    var settings = new CSharpClientGeneratorSettings
    {
        UseBaseUrl = false,
        ClassName = className,
        GenerateClientInterfaces = true,
        CSharpGeneratorSettings =
        {
            Namespace = "HttpClients",
        },
    };

    var generator = new CSharpClientGenerator(document, settings);
    var generatedCode = generator.GenerateFile();

    var path = $"{settings.CSharpGeneratorSettings.Namespace}/{settings.ClassName}";
    var file = new FileInfo(path);
    file.Directory.Create();
    await File.WriteAllTextAsync(file.FullName, generatedCode);
}
```

Using code (NSwag.CodeGeneration.CSharp)



timdeschryver.dev/bits