Test Your ASP.NET Core Web API With Swagger

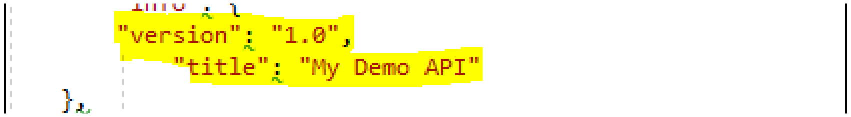
Become a member C# Corner Login
Contribute

SwaggerDemo.zip

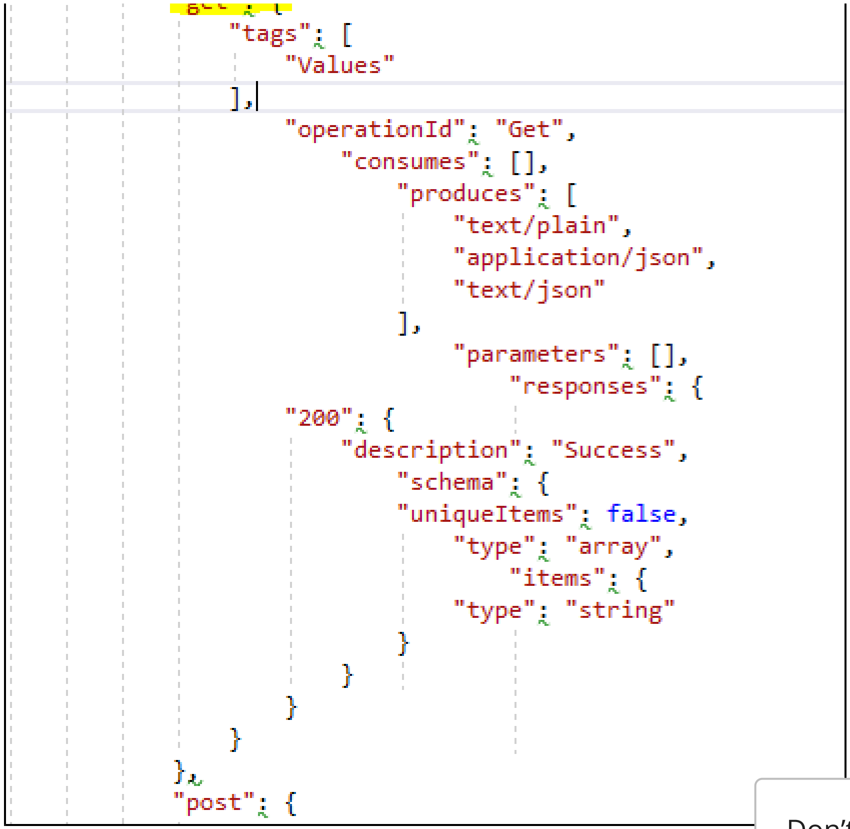
Introduction

Testing of Web APIs is always a challenge because it exposes the end-point rather than the UI. Testing such things may have a dependency on third-party tools, such as fiddler and Post-Man, to Web API endpoints. Swagger can resolve this issue. It provides the UI representation of the RESTful APIs without any implementation logic. It allows the users to understand the capabilities of a service without any code access and it also reduces the amount of time to create a service document.

Swagger generates the UI using the Swagger specification file (swagger.json) that is generated by Swagger tool based on service code. This file describes the capabilities of service; i.e., how many methods are supported by the service and provides information about method parameters. Using this file, Swagger UI generates client coc
swagger.json file.



```
  "info": {
  "version": "1.0",
    "title": "My Demo API"
},
```



```
"get": {
  "tags": [
    "Values"
  ],
    "operationId": "Get",
      "consumes": [],
        "produces": [
          "text/plain",
          "application/json",
          "text/json"
        ],
          "parameters": [],
            "responses": {
    "200": {
      "description": "Success",
        "schema": {
        "uniqueItems": false,
          "type": "array",
            "items": {
          "type": "string"
        }
      }
    }
  }
},
"post": {
```

Swagger can be implemented in ASP.net core web API using Swashbuckle.AspNetCore
source projects to generate Swagger documents for ASP.NET Core Web API. Additiona
to generate TypeScript client code as well as C# servercode for API.

Configure Swagger in ASP.net Core Web API using Swashbuckle.AspNetCore

Step 1 - Install package Swashbuckle.AspNetCore

Using the following command, we can install a Swashbuckle.AspNetCore package.

```
PM> Install-Package Swashbuckle.AspNetCore
```

Step 2 - Configure Swagger middleware

To add Swagger middle to the request pipeline, we need to AddSwaggerGen method in ConfigureService method of startup class. Here, we can define one or more Swagger documents.

Startup.cs

```
public void ConfigureServices(IServiceCollection services)
{
    services.AddMvc().SetCompatibilityVersion(CompatibilityVersion.Version_2_1);
    services.AddSwaggerGen(c =>
    {
        c.SwaggerDoc("v1.0", new Info { Title = "My Demo API", Version = "1.0" });
    });
}
```

If we want to enable this middleware, we need to call UseSwagger method in the configure method of startup class. Here also we need to configure SwaggerEndpoint to generate UI. The UseSwaggerUI is adding a swagger.json file.

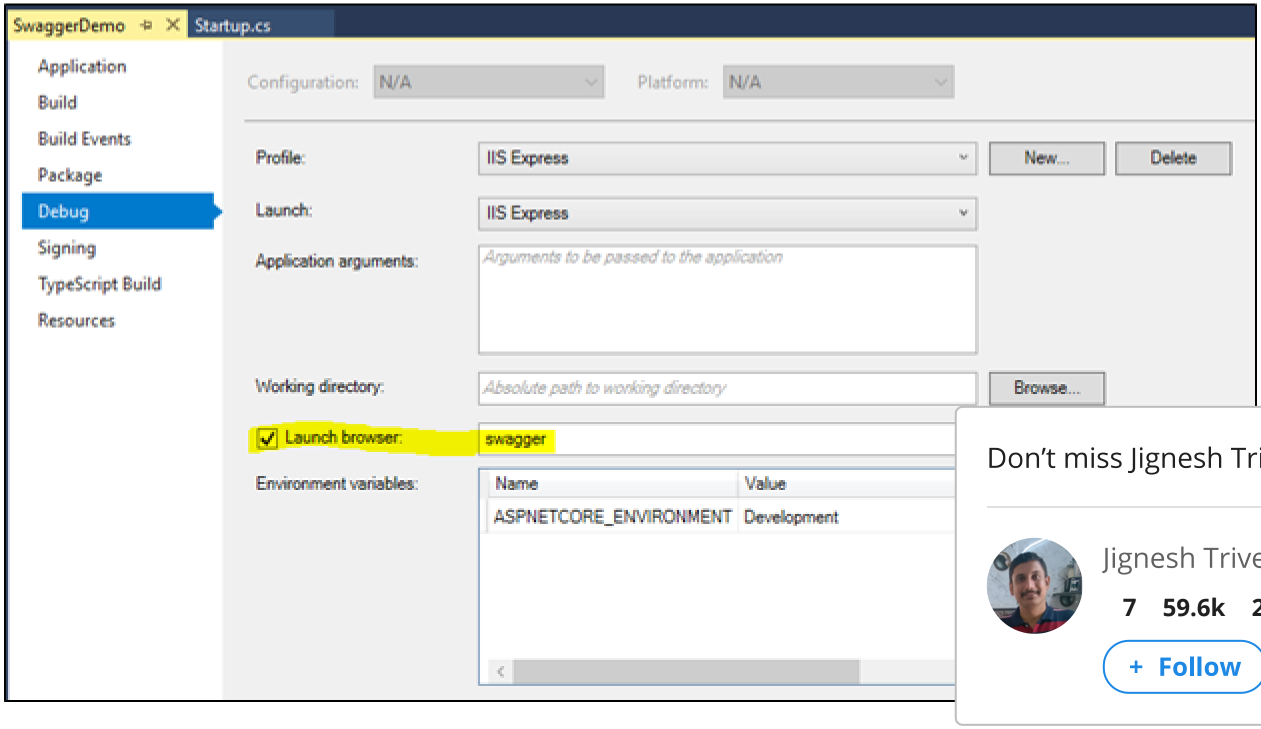
```
public void Configure(IApplicationBuilder app, IHostingEnvironment
{
    if (env.IsDevelopment())
    {
        app.UseDeveloperExceptionPage();
    }
    else
    {
```

```

    app.UseHttpsRedirection();

    app.UseSwagger();
    app.UseSwaggerUI(c =>
    {
        c.SwaggerEndpoint("/swagger/v1.0/swagger.json", "My Demo API (V 1.0)");
    });
    app.UseMvc();
}
```

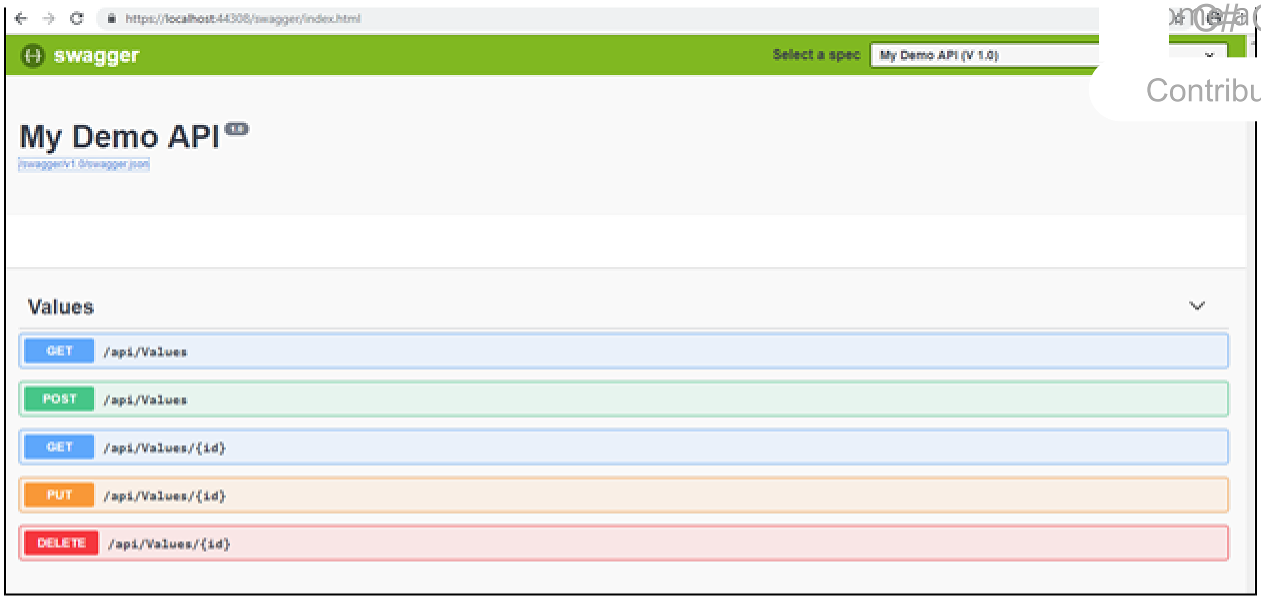
The above steps are required to setup Swagger. If we want to launch Swagger in development environment using Visual Studio, one more change is required. To set Swagger UI, go to project property - debug tab and change Launch Browser value to "swagger".



Configuration: N/A
Platform: N/A
Profile: IIS Express
Launch: IIS Express
Application arguments: Arguments to be passed to the application
Working directory: Absolute path to working directory
☑ Launch browser: swagger
Environment variables:

Name	Value
ASPNETCORE_ENVIRONMENT	Development

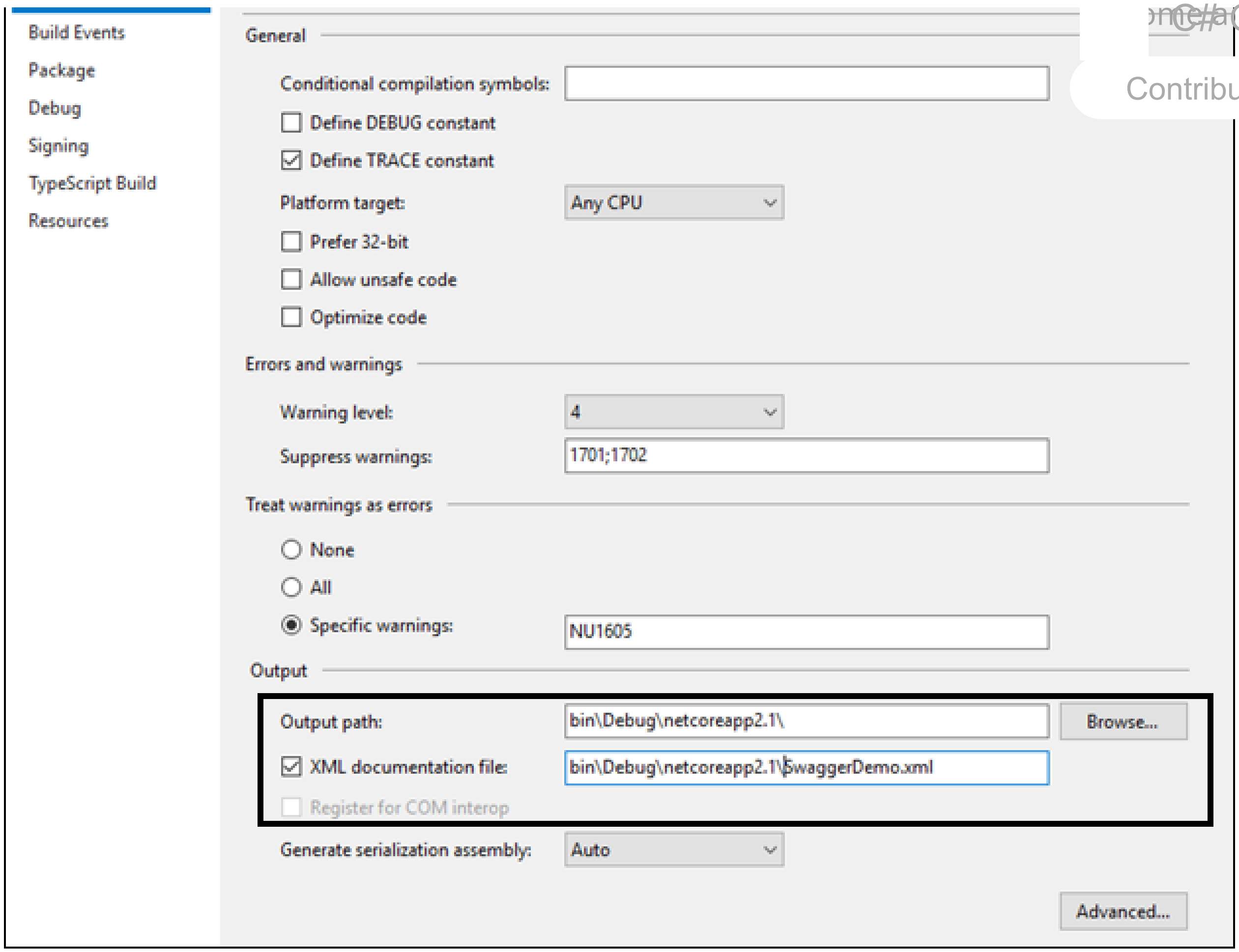
When we run the application, we can see following Swagger UI for ValuesContoller.



As we see here, it uses a different color code for each HTTP verb. When we click on any action method it will ask for parameter details and when we click on excecute button it will send a request to the web API.

The Swagger requires minimum configuration to test our web API. It also shows XML comments when generating UI but that requires some configuration.

The .net /.net core framework provides a way to write XML comment for the documentation. In .net core, we can enble XML comments by setup "XML documentation file" property under build tab of project property windows.



General

Conditional compilation symbols:

☐ Define DEBUG constant

☑ Define TRACE constant

Platform target: Any CPU

☐ Prefer 32-bit

☐ Allow unsafe code

☐ Optimize code

Errors and warnings

Warning level: 4

Suppress warnings: 1701;1702

Treat warnings as errors

○ None

○ All

● Specific warnings: NU1605

Output

Output path: bin\Debug\netcoreapp2.1\

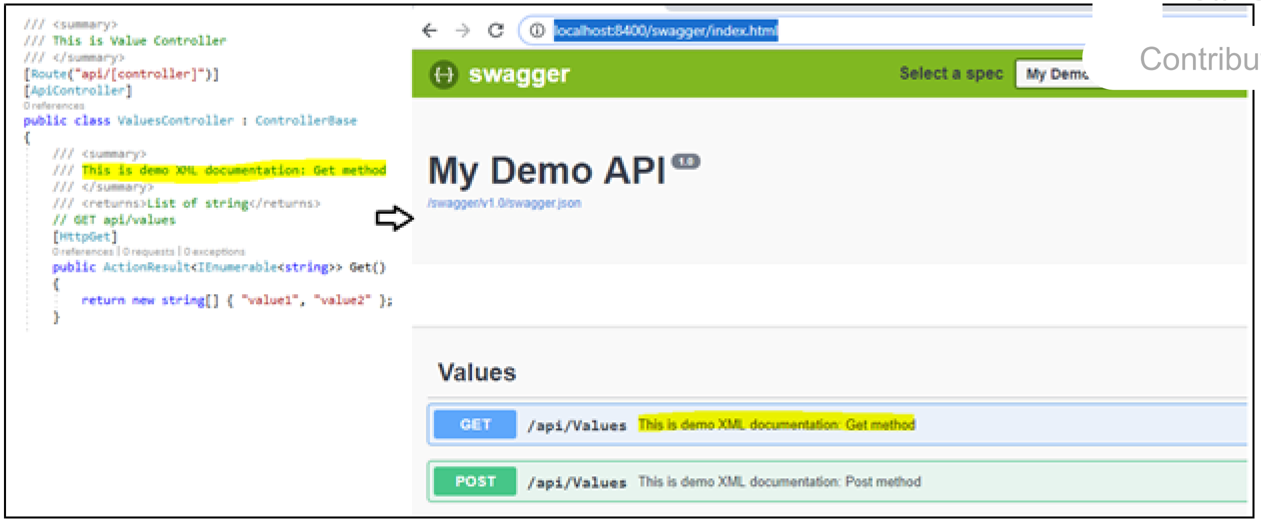
☑ XML documentation file: bin\Debug\netcoreapp2.1\SwaggerDemo.xml

☐ Register for COM interop

Generate serialization assembly: Auto

Swagger UI does not shows this documentation by default. We need to pass the path t

```
public void ConfigureServices(IServiceCollection services)
{
    services.AddMvc().SetCompatibilityVersion(CompatibilityVersion
    services.AddSwaggerGen(c =>
    {
        c.SwaggerDoc("v1.0", new Info { Title = "My Demo API", Version =  "1.0" });
        c.IncludeXmlComments(System.IO.Path.Combine(System.AppContext.BaseDirectory, "SwaggerDemo
    });
```



```
/// <summary>
/// This is Value Controller
/// </summary>
[Route("api/[controller]")]
[ApiController]
0 references
public class ValuesController : ControllerBase
{
    /// <summary>
    /// This is demo XML documentation: Get method
    /// </summary>
    /// <returns>List of string</returns>
    // GET api/values
    [HttpGet]
    0 references | 0 requests | 0 exceptions
    public ActionResult<IEnumerable<string>> Get()
    {
        return new string[] { "value1", "value2" };
    }
```

localhost:8400/swagger/index.html

My Demo API 1.0

/swagger/v1.0/swagger.json

Values

GET /api/Values This is demo XML documentation: Get method

POST /api/Values This is demo XML documentation: Post method

Configure Swagger in ASP.net Core Web API using NSwag

Following are the step to configure Swagger in ASP.net Core Web API using NSwag.AspNetCore

Step 1 - Install-Package NSwag.AspNetCore

Using the following command, we can install package of NSwag.AspNetCore

```
01. PM> Install-Package NSwag.AspNetCore
```

Step 2 - Configure Swagger middleware

To add a Swagger middle to the request pipeline, we need to AddSwaggerDocument n startup class. This method adds default document with name "V1". If we want to add r method multiple times with a different document name.

Startup.cs

```
    services.AddMvc().SetCompatibilityVersion(CompatibilityVersion.Version_    );
    services.AddSwaggerDocument();
}
```

If want to enable this middleware, we need to call UseSwagger method in configure method of startup class. This method adds default route as "/swagger/{documentName}/swagger.json" Also we need to call UseSwaggerUi3 method to add the Swagger UI to request pipeline.

```
public void Configure(IApplicationBuilder app, IHostingEnvironment env)
{
    if (env.IsDevelopment())
    {
        app.UseDeveloperExceptionPage();
    }
    else
    {
        app.UseHsts();
    }

    app.UseHttpsRedirection();

    app.UseSwagger();
    app.UseSwaggerUi3();

    app.UseMvc();
}
```

The above steps are required to set up Swagger. If we want to launch Swagger in deve Studio, one more change is required. To set Swagger UI, go to project property - debu to "swagger".

Test your ASP.NET Core web API with Swagg

Summary

You can view or download the source code from the GitHub link here.

ASP.NET Core ASP.NET Core Web API Swagger Test Your ASP.NET Core Web API

Jignesh Trivedi TOP 50

Jignesh Trivedi is a Developer, C# Corner MVP, Microsoft MVP, Author, Blogger,eager to learn new technologies

https://www.c-sharpcorner.com/members/jignesh-trivedi

7 23.2m 7 2

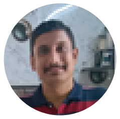
10 12

Type your comment here and press Enter Key (Minimum 10 characters)

How to send authenticated request using swashbuckle ?

shrey shah Aug 12, 2019

1815 29 0 0 1 Reply



You have to configure AddSecurityDefinition and AddSecurityRequirement me
done you can find Authorize button at the top right of the page.

Jignesh Trivedi

7 59.6k 23.2m

Don’t miss Jignesh Trivedi’s next article ×

Jignesh Trivedi

7 59.6k 23.2m

+ Follow

How to generate server side dotnet core api code using visual studio code from yaml file

Koteswara Rao

1823 21 0

Thank you. This helps a lot.

Facing the following issue "HTTP ERROR 404" while accessing "https://localhost:44347/swagger"

Mushtaq M A Jun 05, 2019

234 8k 714.2k 0 2 Reply

I hope you have done all the configuration define in this article. Can you please try "index.html" at the end of your URL

Jignesh Trivedi Jun 05, 2019

7 59.6k 23.2m 0

I have accessed as you mentioned "https://localhost:44347/swagger/index.html" and displays blanks screen "http://www.mediafire.com/view/3eg9cfcoo49ncq4/swagger_ui.png"

Mushtaq M A Jun 05, 2019

234 8k 714.2k 0

Good example.......Thanks

Hamid Khan Mar 07, 2019

621 2.5k 241.4k 1 0 Reply

Great tutorial.

Nelson Giraldo Feb 26, 2019

1836 8 0 1 0 Reply

Hi Jignesh, Is there any way by which we can monitor(check whether API is up or down) our ASP.NET web API's using swagger without using any third party tools? Swagger provides a tool called AlertSite which help in monitoring API's but I cannot use a third party tool. Is there anything which Nuget can help with?

Sagar Sinha

1819 25 0

Don't miss Jignesh Trivedi's next article ×

Jignesh Trivedi

7 59.6k 23.2m

+ Follow

Excellent usefull article...

Manish Patel

1841 3 0

Very good concept explain... Thanks Jignesh

Rushi Mehta Dec 1(

C# Corner Conference 2020 Dates Announced

ome a member C# Corner Login

Contribute

TRENDING UP

01 Learn Angular 8 Step By Step in 10 Days – Day 1

02 Learn Angular 8 Step By Step in 10 Days – Components (Day 2)

03 ASP.NET Core And Blazor Code Venture - Configuring Azure AD Authentication

04 Overview Of Session In ASP.NET

05 Execute A Stored Procedure Programmatically

06 T-SQL For Beginners

07 A Look Into Angular Material Design And Fixing Issues During Installation🔧If Any

08 Static Class And Static Class Members In C#

09 Preparing And Mastering The Microsoft Azure AZ-300 And AZ-301 Exams

10 Make SharePoint Online Batch API Request Easy In Single Line Of Code

View All ○

Don't miss Jignesh Trivedi's next article ×

Jignesh Trivedi

7 59.6k 23.2m

+ Follow

About Us Contact Us Privacy Policy Terms Media Kit Sitemap Repo

C# Tutorials Common Interview Questions