**Introducción**

Ya he escrito en este blog en alguna ocasión sobre **Swagger** en ASP.NET Core, sin embargo, en los últimos meses **Swagger** ha sufrido algunos cambios y actualizaciones que pueden hacerte perder la cabeza a la hora de aplicarlo a nuestras Web APIs con ASP.NET Core 3.1.

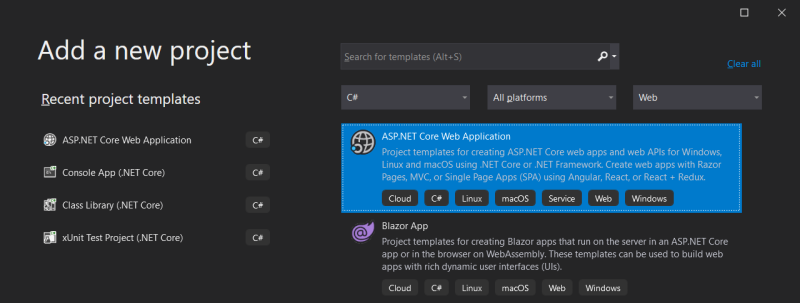
Mi intención en esta entrada es la de explicarte cómo aplicar en tres simples pasos el uso de **Swagger** a nuestras Web APIs y no morir en el camino.

Para el que no lo sepa, **Swagger** nos permite documentar nuestra Web API facilitándonos información de entrada, salida, parámetros, etc., e incluso probar nuestra Web API.

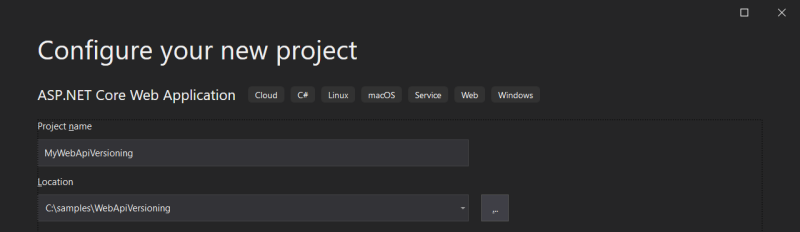
**Crear nuestro proyecto**

En primer lugar, empezaremos creando nuestro proyecto de Web API.  
Voy a utilizar la base que se crea directamente al crear una nueva Web API en Visual Studio 2019.

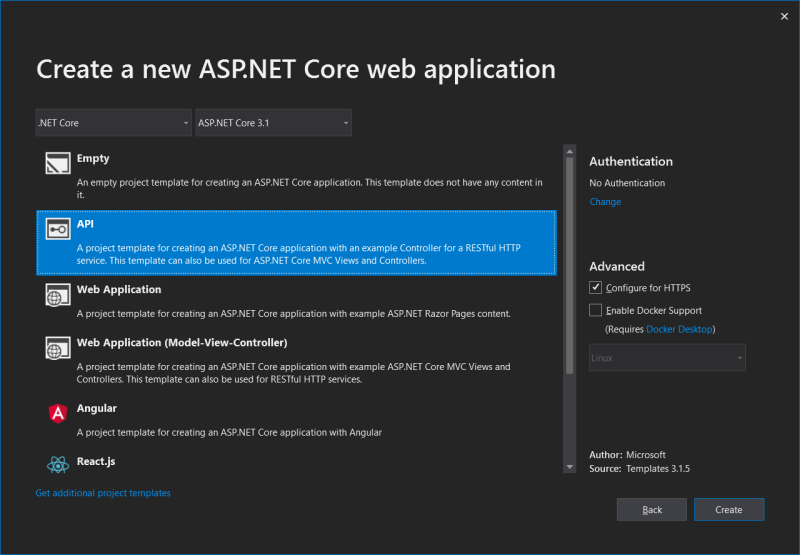
Seleccionaremos como tipo de proyecto **ASP.NET Core Web Application**.

[](https://geeks.ms/jorge/wp-content/uploads/sites/6/2020/06/20200602_01.png)

Configuraremos el proyecto:

[](https://geeks.ms/jorge/wp-content/uploads/sites/6/2020/06/20200602_02.png)

Y seleccionaremos el tipo de plantilla para nuestro proyecto de **ASP.NET Core Web Application**, que en nuestro caso será **API**.

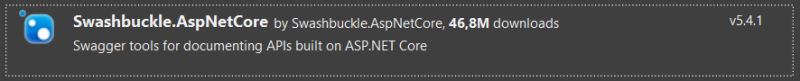
[](https://geeks.ms/jorge/wp-content/uploads/sites/6/2020/06/20200602_03.png)

Nuestro proyecto tiene un controlador por defecto denominado ***WeatherForecastController***, y sobre él voy a aplicar la documentación de **Swagger**.

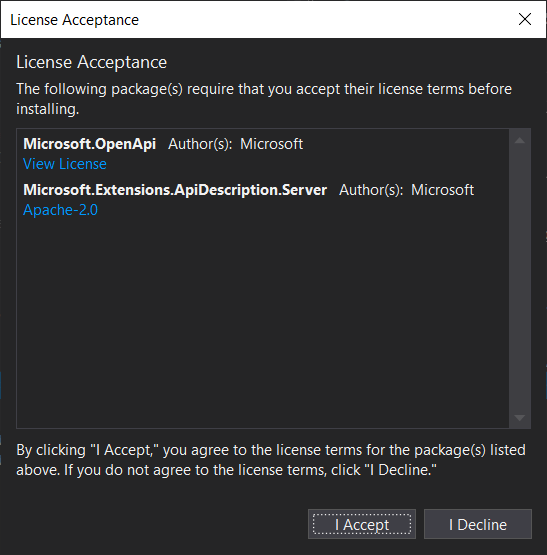
**Agregando Swagger**

Para poder utilizar **Swagger** en nuestra Web API, deberemos utilizar un paquete NuGet de nombre **Swashbuckle.AspNetCore**.

Este paquete puede ser localizado [en NuGet](https://www.nuget.org/packages/Swashbuckle.AspNetCore).

[](https://geeks.ms/jorge/wp-content/uploads/sites/6/2020/06/20200602_04.png)

Instalaremos el paquete, que a su vez nos pedirá que aceptemos la licencia de uso sobre otras dos fuentes.

[](https://geeks.ms/jorge/wp-content/uploads/sites/6/2020/06/20200602_05.png)

Y con esto, estamos preparados para empezar a utilizar **Swagger** dentro de nuestra Web API.  
Lo único que debemos hacer ahora, es configurar **Swagger** en nuestro proyecto.

**Configurando Swagger en nuestra Web API de ASP.NET Core 3.1**

Abriremos el fichero **Startup.cs**.

Para que **Swagger** funcione correctamente, deberemos agregar un **using** a nuestro código que nos permita trabajar con los modelos de **OpenApi**.

using Microsoft.OpenApi.Models;

Dentro del fichero **Startup.cs** nos posicionaremos dentro del método **ConfigureServices**.

Dentro de este método y por hacerlo fácil, agregaremos el siguiente código:

public void ConfigureServices(IServiceCollection services)

{

services.AddControllers();

**AddSwagger(services);**

}

**private void AddSwagger(IServiceCollection services)**

**{**

**services.AddSwaggerGen(options =>**

**{**

**var groupName = "v1";**

**options.SwaggerDoc(groupName, new OpenApiInfo**

**{**

**Title = $"Foo {groupName}",**

**Version = groupName,**

**Description = "Foo API",**

**Contact = new OpenApiContact**

**{**

**Name = "Foo Company",**

**Email = string.Empty,**

**Url = new Uri("https://foo.com/"),**

**}**

**});**

**});**

}

También nos posicionaremos en el método **Configure** y agregaremos el siguiente código:

public void Configure(IApplicationBuilder app, IWebHostEnvironment env)

{

if (env.IsDevelopment())

{

app.UseDeveloperExceptionPage();

}

app.UseHttpsRedirection();

**app.UseSwagger();**

**app.UseSwaggerUI(c =>**

**{**

**c.SwaggerEndpoint("/swagger/v1/swagger.json", "Foo API V1");**

**});**

app.UseRouting();

app.UseAuthorization();

app.UseEndpoints(endpoints =>

{

endpoints.MapControllers();

});

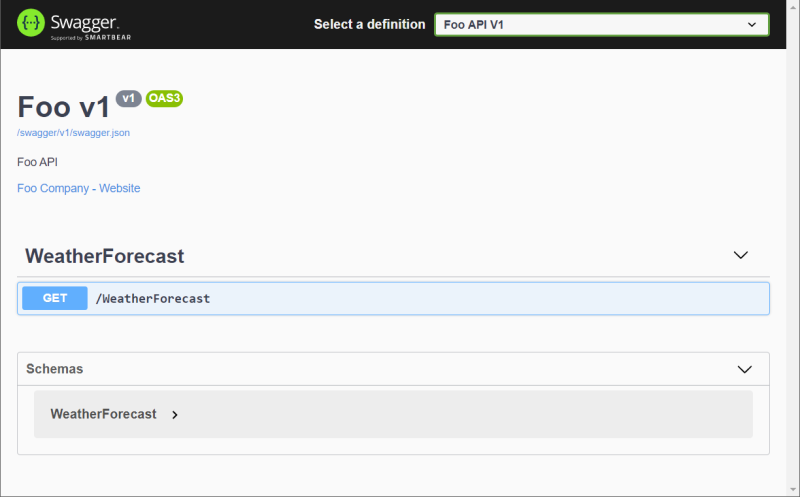
}

**Accediendo a la documentación de nuestra Web API**

Ya estamos listos para comprobar la documentación de nuestra Web API.

Ejecutaremos nuestra Web API desde Visual Studio 2019 y acudiremos a la ruta ***https://localhost:1234/swagger*** (dónde 1234 será el puerto de nuestra Web API).

Tendremos que ver algo parecido a lo siguiente:

[](https://geeks.ms/jorge/wp-content/uploads/sites/6/2020/06/20200602_06.png)