UD7 Servicios web.

7.3 Documentación de la API.

A la hora de consumir un servicio web, es importante para el desarrollador conocer los métodos disponibles y los parámetros que necesitan. Swagger, también conocido como OpenAPI, es una herramienta que ayuda a solucionar el problema de generar documentación y páginas de ayuda para APIs web.

Existen diversas implementaciones de Swagger para ASP.NET, siendo las más conocidas Swashbuckle.AspNetCore y NSwag. En esta parte de la unidad veremos como generar la documentación para nuestro servicio web utilizando Swashbuckle.AspNetCore, que es un proyecto open source para generar documentación Swagger para proyectos ADP.NET Core Web API.

Swagger / OpenAPI.

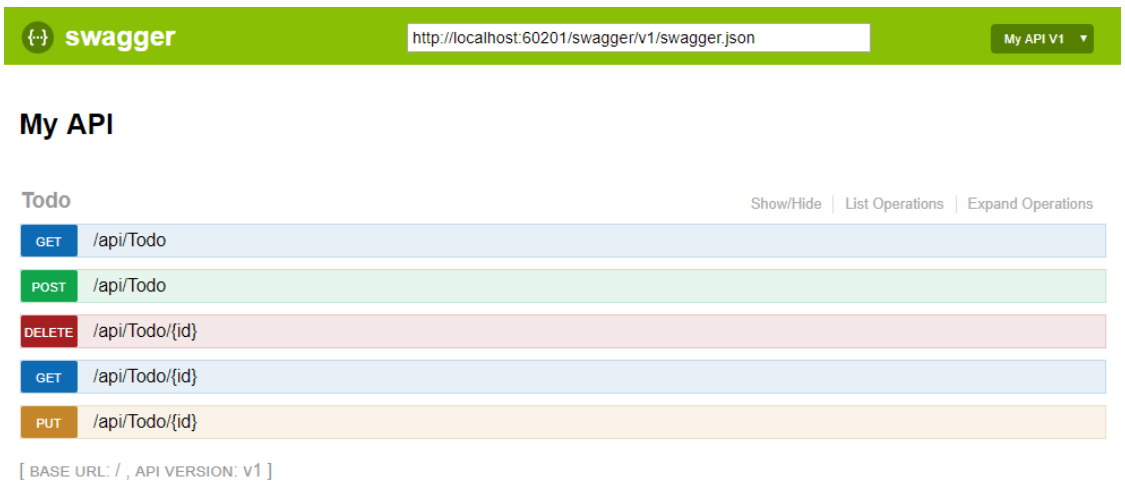
Swagger es una especificación independiente del lenguaje para describir APIs REST. El proyecto Swagger fue donado a la OpenAPI Initiative, pasando a denominarse OpenAPI. Permite que tanto sistemas como personas puedan obtener información sobre un servicio sin necesidad de conocer la implementación. Su objetivo es minimizar la cantidad de trabajo necesaria para utilizar el servicio y generar la documentación relativa al mismo.

El elemento central es la llamada especificación Swagger. Se trata de un fichero denominado swagger.json. Este documento es generado por las herramientas de OpenAPI o alguna de las implementaciones de terceros, a partir de nuestro servicio. Este documento es empleado por Swagger UI para ofrecer la documentación del servicio y permite la generación de código cliente.

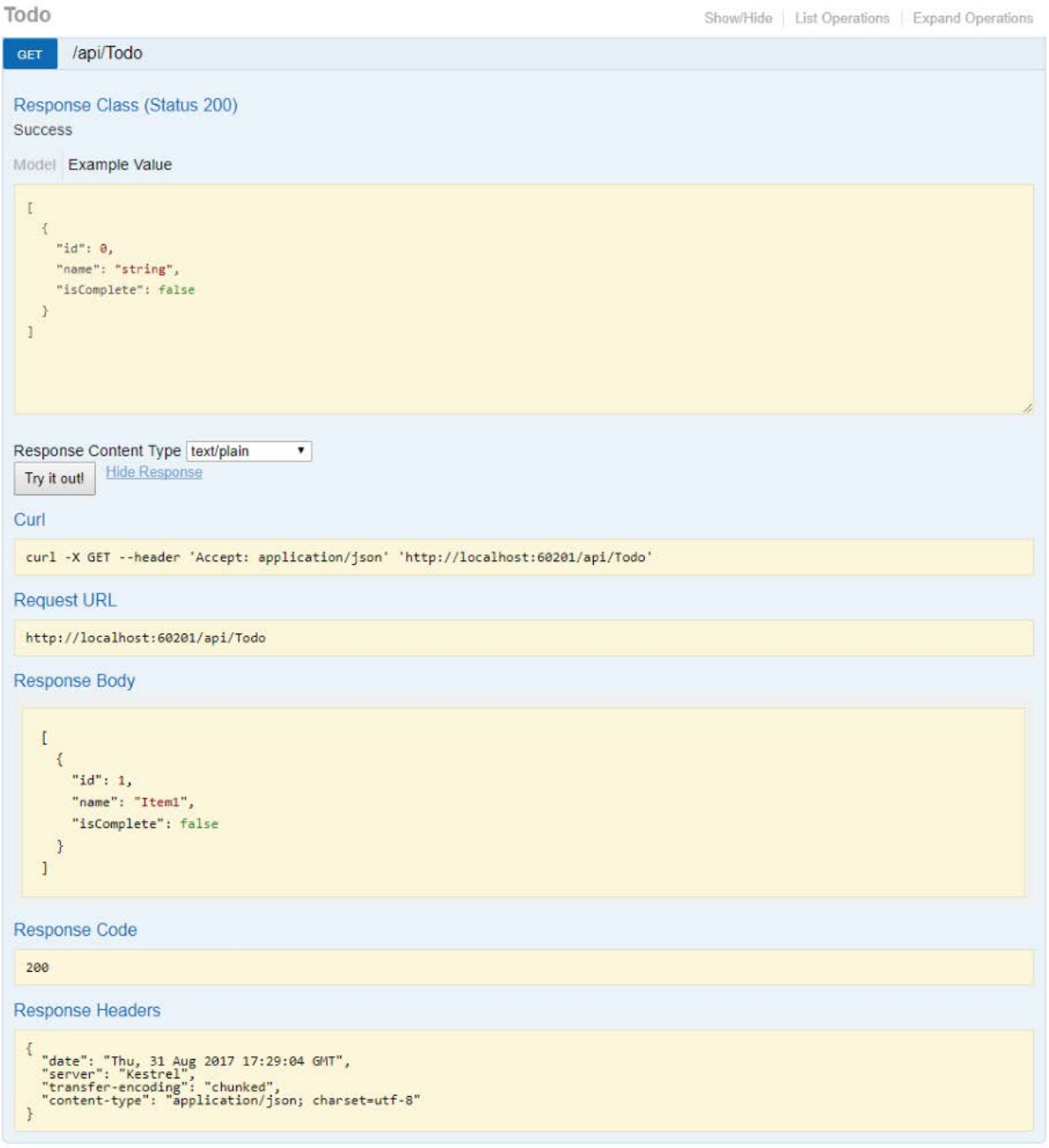
Swagger UI.

Swagger UI ofrece una UI web que proporciona toda la información sobre nuestro servicio, generada a partir de la especificación Swagger. Tanto Swashbuckle como NSwag inluyen una versión de Swagger UI, que permiten hostear la documentación web generada en una aplicación ASP.NET Core.

La documentación generada tiene una apariencia similar a la siguiente:



La UI permite también probar cada uno de los métodos de la API. Para ello haremos clic en el nombre de un método, añadiremos los parámetros necesarios y le daremos a probar.

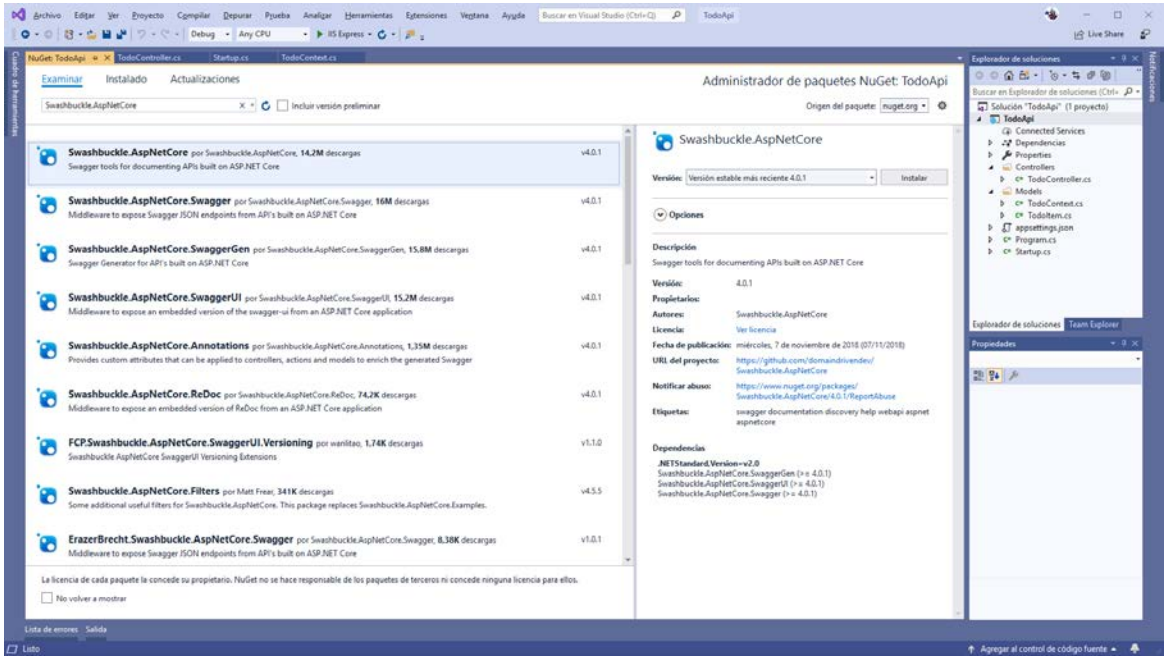


CIFP A Carballeira-Marcos Valcárcel
Rúa A Carballeira s/n
Ourense CP 32002
Tfno. 988788470, Fax 988227935

Ejemplo: Generación de la documentación para el servicio con Swashbuckle.

1. Abrimos el proyecto Web API creado en esta unidad.

2. Instalamos paquete Swashbuckle.AspNetCore desde el gestor de paquetes Nuget:



3. Añadimos el generador de Swagger a la colección de servicios de la aplicación en *ConfigureServices*:

```
public void ConfigureServices(IServiceCollection services)
{
    services.AddDbContext<TodoContext>(opt =>
        opt.UseInMemoryDatabase("TodoList"));

    services.AddMvc().SetCompatibilityVersion(CompatibilityVersion.Version_2_2);

    // Register the Swagger generator, defining 1 or more Swagger documents
    services.AddSwaggerGen(c =>
    {
        c.SwaggerDoc("v1", new Info { Title = "Todo API", Version = "v1" });
    });
}
```

4. Habilitamos el módulo que sirve el fichero JSON generado y la Swagger UI, en *Configure*:

```
// This method gets called by the runtime. Use this method to configure the HTTP
request pipeline.
public void Configure(IApplicationBuilder app, IHostingEnvironment env)
{
    if (env.IsDevelopment())
    {
        app.UseDeveloperExceptionPage();
    }
    else
```

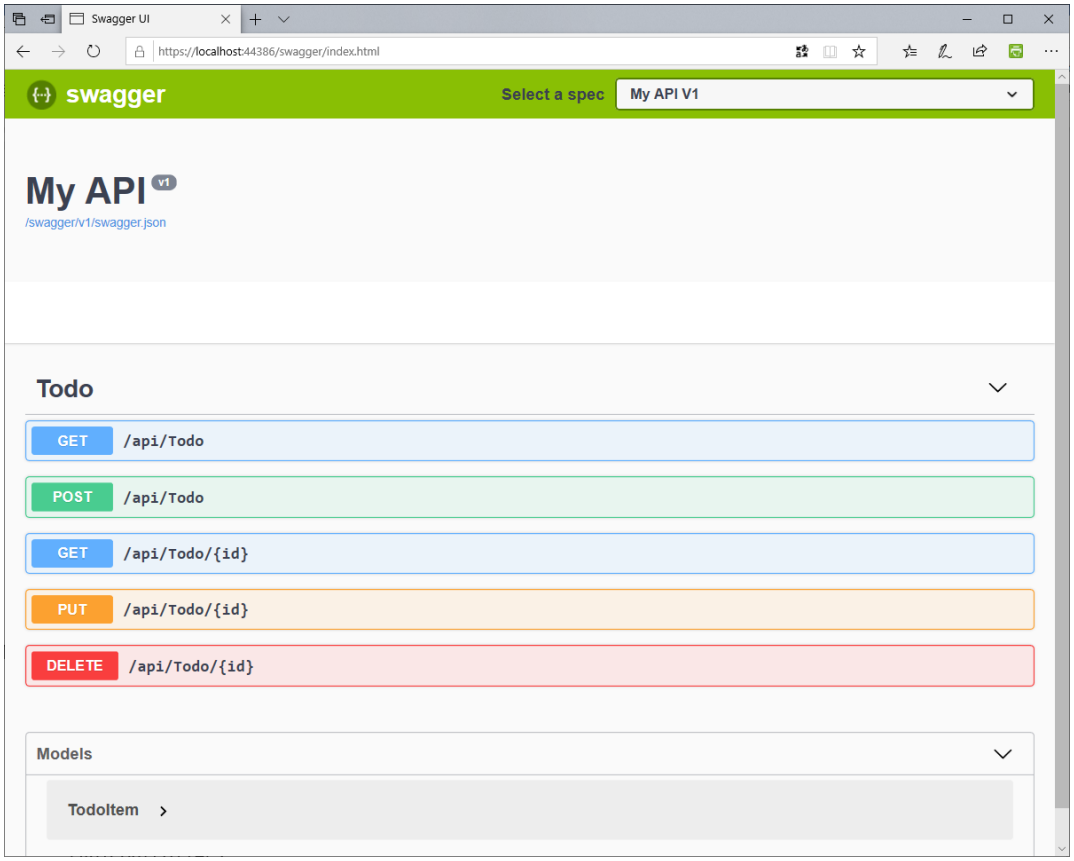
```
    {
    // The default HSTS value is 30 days. You may want to change this for production
scenarios, see https://aka.ms/aspnetcore-hsts.
        app.UseHsts();
    }

    // Enable middleware to serve generated Swagger as a JSON endpoint.
    app.UseSwagger();

    // Enable middleware to serve swagger-ui (HTML, JS, CSS, etc.),
    // specifying the Swagger JSON endpoint.
    app.UseSwaggerUI(c =>
    {
        c.SwaggerEndpoint("/swagger/v1/swagger.json", "My API V1");
    });

    app.UseHttpsRedirection();
    app.UseMvc();
}
```

5. Ejecutamos nuestra aplicación Web API, y navegamos a https://localhost:xxxxx/swagger y podremos ver la documentación generada de nuestra API, incluyendo todos los métodos con sus URLs correspondientes y los modelos utilizados.



Como hemos podido comprobar, generar la documentación para nuestras APIs es extremadamente sencillo gracias a Swashbuckle. Para más información acerca de la personalización de la documentación generada y el uso de la herramienta se puede consultar en la web del proyecto:

https://github.com/domaindrivendev/Swashbuckle.AspNetCore