



A <PROGRAMAÇÃO> PRECISA DE TI

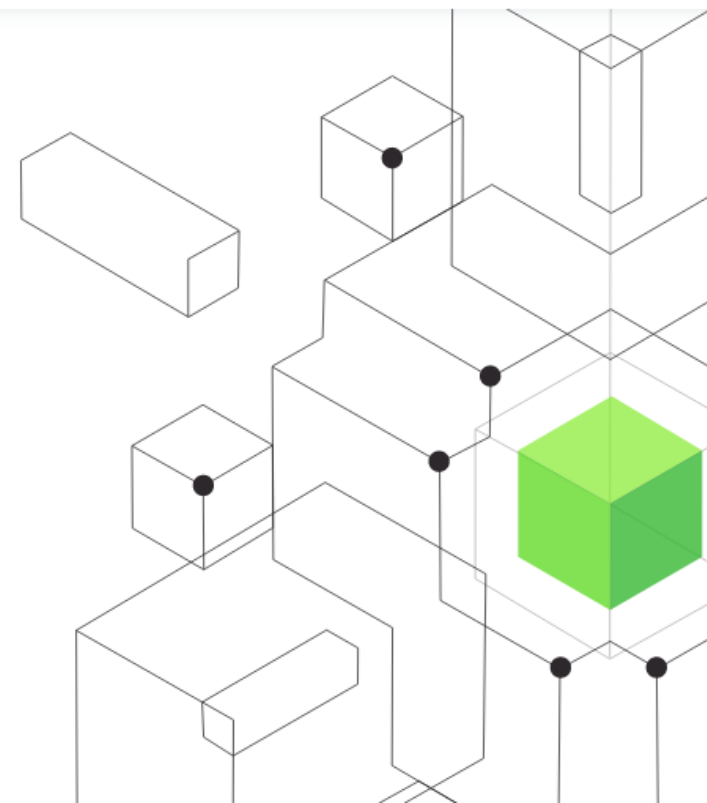
Swagger & Package Manager

Laboratório Web



API Development for Everyone

Simplify API development for users, teams, and enterprises with the Swagger open source and professional toolset. Find out how Swagger can help you design and document your APIs at scale.

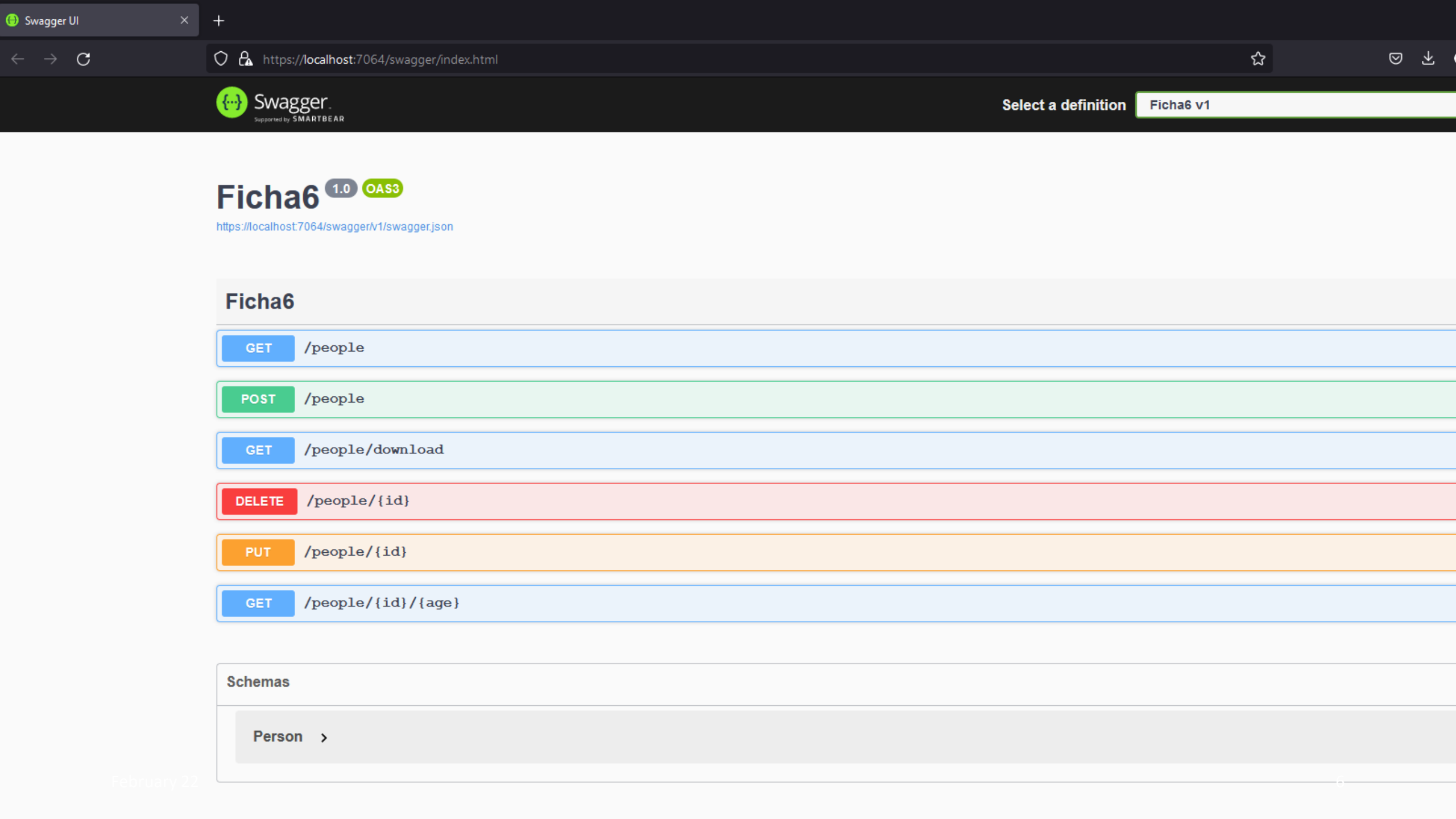


Swagger

- Swagger é uma Interface Description Language para descrever APIs RESTful (<https://swagger.io/specification/>)
- Swagger é utilizado em conjunto com outras ferramentas para desenhar, implementar e testar serviços web RESTful
- É baseado num standard da indústria denominado por *OpenAPI Specification* que é a especificação padrão para o desenho de APIs RESTful

```
"/pets": {  
  "get": {  
    "description": "Returns all pets from the system that the user has access to",  
    "operationId": "findPets",  
    "produces": [  
      "application/json",  
      "application/xml",  
      "text/xml",  
      "text/html"  
    ],  
    "parameters": [  
      {  
        "name": "tags",  
        "in": "query",  
        "description": "tags to filter by",  
        "required": false,  
        "type": "array",  
        "items": {  
          "type": "string"  
        }  
      }  
    ]  
  }  
}
```

```
    "name": "limit",
    "in": "query",
    "description": "maximum number of results to return",
    "required": false,
    "type": "integer",
    "format": "int32"
  }
],
"responses": {
  "200": {
    "description": "pet response",
    "schema": {
      "type": "array",
      "items": {
```



Ficha6

1.0

OAS3

<https://localhost:7064/swagger/v1/swagger.json>

Ficha6

GET /people

POST /people

GET /people/download

DELETE /people/{id}

PUT /people/{id}

GET /people/{id}/{age}

Schemas

Person >

Ficha6



GET

/people

^

Parameters

Try it out

No parameters

Responses

Code	Description	Links
200	Success	No links

```
curl -X 'GET' \  
  'https://localhost:7064/people' \  
  -H 'accept: */*'
```

Request URL

https://localhost:7064/people

Server response

Code

Details

200

Response body

```
{  
  "personList": [  
    {  
      "firstName": "Pedro",  
      "lastName": "Lopes",  
      "profession": "Teacher",  
      "age": 22,  
      "id": 1  
    },  
    {  
      "firstName": "Ana",  
      "lastName": "Matos",  
      "profession": "Driver",  
      "age": 28,  
      "id": 2  
    },  
    {  
      "firstName": "Marco",  
      "lastName": "Gouda",  
      "profession": "Student",  
      "age": 17,  
      "id": 3  
    }  
  ]  
}
```



Download

Response headers

```
content-type: application/json; charset=utf-8  
date: Sat, 05 Feb 2022 21:05:42 GMT  
server: Kestrel  
x-ffox-spdy: h2
```

Responses

February 22

8

Code

Description

Links

Como criar um projeto com Swagger?

Additional information

ASP.NET Core Web API

C#

Linux

macOS

Windows

Cloud

Service

Web

Framework ⓘ

.NET 6.0 (Long-term support)

Authentication type ⓘ

None

☒ Configure for HTTPS ⓘ

☐ Enable Docker ⓘ

Docker OS ⓘ

Linux

☐ Use controllers (uncheck to use minimal APIs) ⓘ

☒ Enable OpenAPI support ⓘ

Como ativar o Swagger num projeto já criado?

1. Instalar a dependência necessária através do Package Manager (já lá vamos)
2. Adicionar o Swagger generator à lista de serviços do nosso builder
3. Ativar o middleware para servir o documento JSON gerado pelo Swagger
4. Ativar o middleware para a Swagger UI

Como ativar o Swagger num projeto já criado?

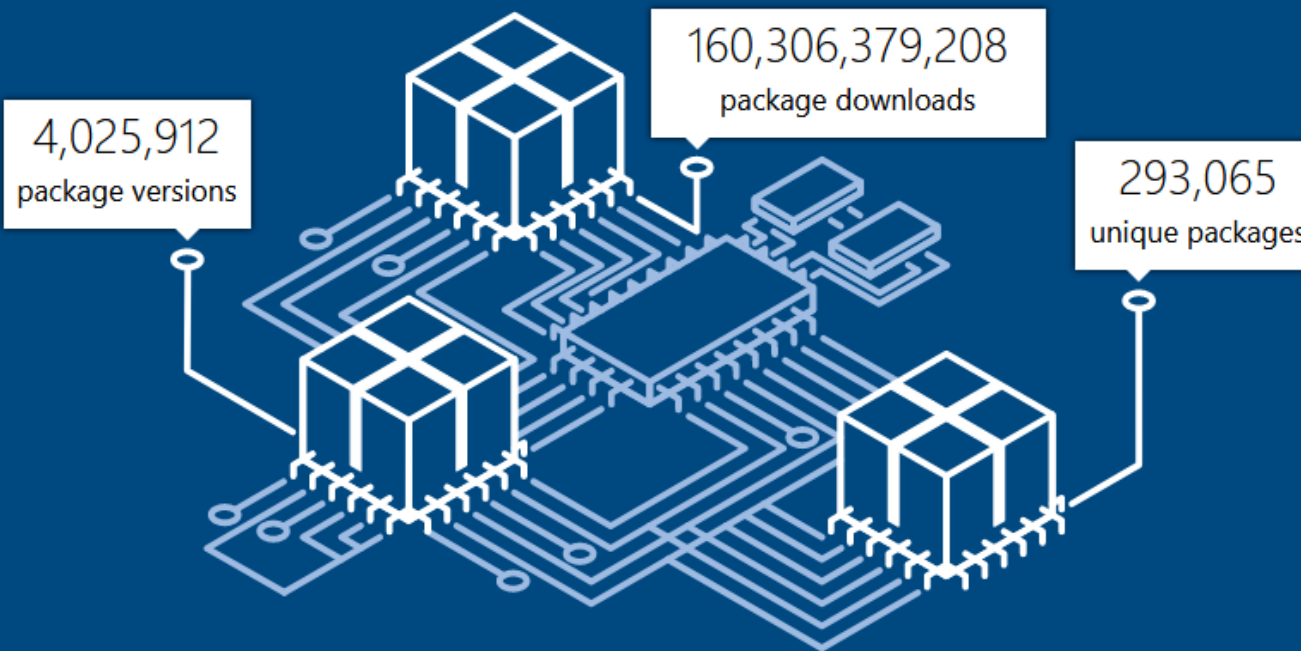
```
// Add services to the container.  
// Learn more about configuring Swagger/OpenAPI at https://aka.ms/aspnetcore/swashvortex  
builder.Services.AddEndpointsApiExplorer();  
builder.Services.AddSwaggerGen();
```

```
// Configure the HTTP request pipeline.  
if (app.Environment.IsDevelopment())  
{  
    app.UseSwagger();  
    app.UseSwaggerUI();  
}
```

- ApiExplorer permite expôr os nossos endpoints, os seus URLs, métodos HTTP, parâmetros e tipos de respostas
- Ativamos o Swagger e o SwaggerUI como middleware

Create .NET apps faster with NuGet

Search for packages...



NUGET

- É o gestor de pacotes para o .NET
- Permite produzir e consumir pacotes
- Estes pacotes estão disponíveis na galeria do NuGet que funciona como repositório central usado pelos autores e consumidores
- Pode ser utilizado de várias formas
 - dotnet CLI
 - Package manager UI
 - Package manager console

O que é um pacote?

- De grosso modo é um ficheiro ZIP com a extensão **.nupkg**
- Esse ficheiro contém todo o código compilado do nosso projeto e outros ficheiros que sejam necessários
- Também contém um ficheiro (**manifest**) que descreve o pacote e a versão do mesmo
- Ao instalarmos esse pacote, podemos utilizar as suas funcionalidades

dotnet CLI

- Abrir a linha de comandos do Windows
- Navegar para a pasta onde está a nossa solução/projeto (**cd ...**)
- Listar todos os pacotes instalados
 - **dotnet** list package
- Instalar um pacote
 - **dotnet** add package Newtonsoft.Json --version 12.0.1

Package manager UI

- No Visual Studio procurar por “**Manage Nuget Packages**”
- É possível consultar os pacotes instalados e se existem atualizações
- Também é possível pesquisar por pacotes e instalar os mesmos
- Os pacotes apresentados estão no repositório central do NuGet

Browse

Installed

Updates

NuGet Package Manager: Fich6

swagger



Include prerelease

Package source: nuget.org



Swashbuckle.AspNetCore.Swagger by Swashbuckle.AspNetCore.Swagger, **191M** downloads 6.2.3
Middleware to expose Swagger JSON endpoints from APIs built on ASP.NET Core



Swashbuckle.AspNetCore.SwaggerGen by Swashbuckle.AspNetCore.SwaggerGen, **189M** downlo 6.2.3
Swagger Generator for APIs built on ASP.NET Core



Swashbuckle.AspNetCore.SwaggerUI by Swashbuckle.AspNetCore.SwaggerUI, **174M** downloads 6.2.3
Middleware to expose an embedded version of the swagger-ui from an ASP.NET Core application



Swashbuckle.AspNetCore by Swashbuckle.AspNetCore, **158M** downloads 6.2.3
Swagger tools for documenting APIs built on ASP.NET Core



Microsoft.Extensions.ApiDescription.Server by Microsoft, **109M** downloads 6.0.1
MSBuild tasks and targets for build-time Swagger and OpenApi document generation



NSwag.SwaggerGeneration by Rico Suter, **5,45M** downloads 12.3.0
NSwag: The Swagger API toolchain for .NET and TypeScript



Swagger by Ryan, **174K** downloads 1.0.0
Provides a Swagger object model and middleware that can be used to expose Swagger descriptions for your API.

Each package is licensed to you by its owner. NuGet is not responsible for, nor does it grant any licenses to, third-party packages.

☐ Do not show this again

Swashbuckle.AspNetCore

Installed: 6.2.3

Uninstall

Version: 6.2.3

Update

Options

Description

Swagger tools for documenting APIs built on ASP.NET Core

Version: 6.2.3

Author(s): Swashbuckle.AspNetCore

License: MIT

Downloads: 158,493,983

Date published: Saturday, October 16, 2021 (10/16/2021)

Project URL: <https://github.com/domaindrivendev/Swashbuckle.AspNetCore>

Report Abuse: <https://www.nuget.org/packages/Swashbuckle.AspNetCore/6.2.3/ReportAbuse>

Tags: swagger, documentation, discovery, help, webapi, aspnet, aspnetcore

Dependencies

└─ .NETStandard,Version=v2.0

Swashbuckle.AspNetCore.Swagger (>= 6.2.3)

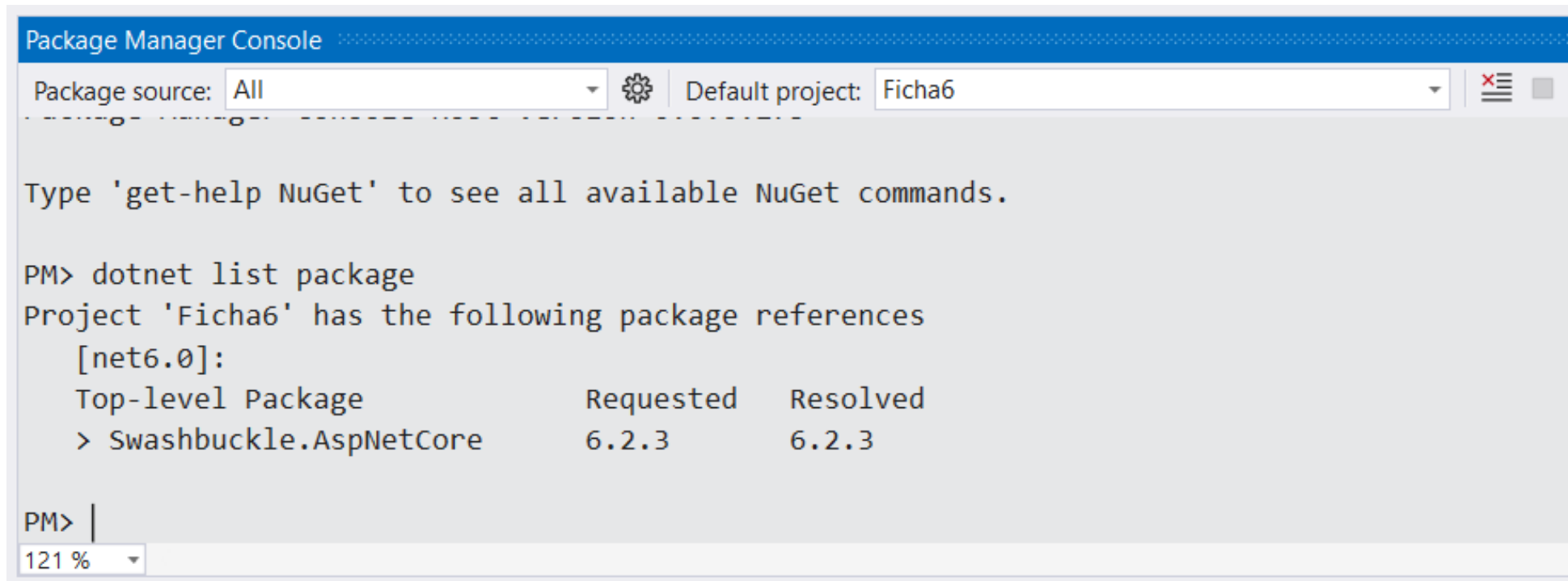
Swashbuckle.AspNetCore.SwaggerGen (>= 6.2.3)

Swashbuckle.AspNetCore.SwaggerUI (>= 6.2.3)

Microsoft.Extensions.ApiDescription.Server (>= 3.0.0)

Package manager console

- Tem as mesmas funcionalidades que as opções anteriores



```
Package Manager Console
Package source: All [gear icon] Default project: Ficha6 [dropdown icon] [close icon] [maximize icon]

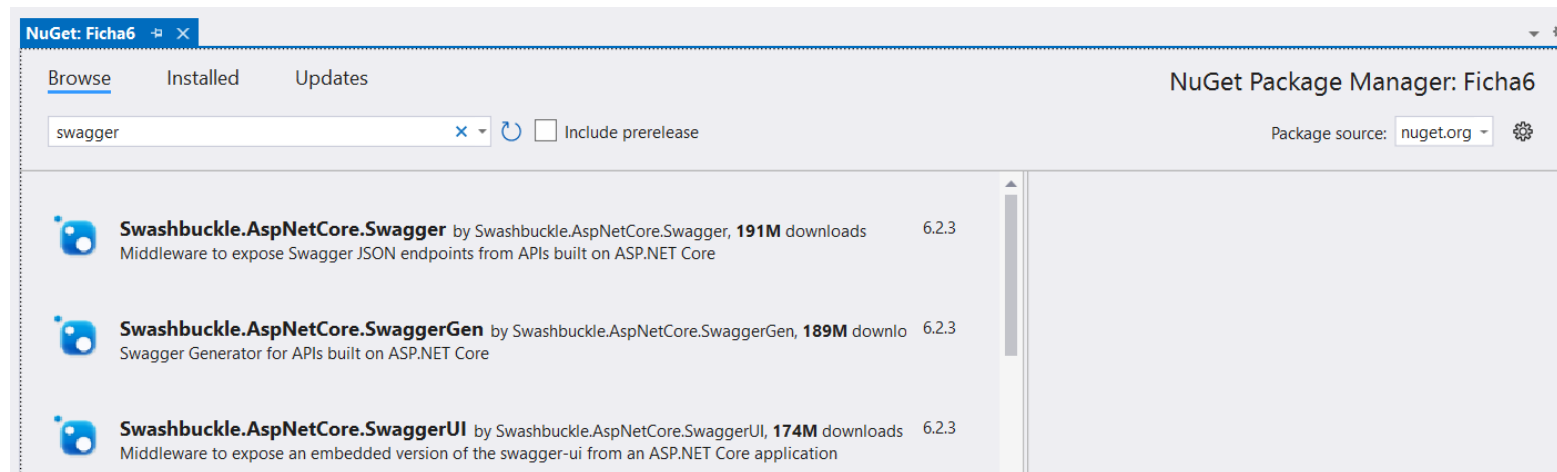
Type 'get-help NuGet' to see all available NuGet commands.

PM> dotnet list package
Project 'Ficha6' has the following package references
  [net6.0]:
  Top-level Package           Requested  Resolved
  > Swashbuckle.AspNetCore    6.2.3     6.2.3

PM> |
121 %
```

Como instalar o Swagger?

- Pode ser instalado através de qualquer uma das opções anteriores
- O mais fácil é procurar pelo nome no Package manager UI
- Selecionar o pacote e efetuar a instalação



Referências

The screenshot shows a web browser displaying the Microsoft documentation page for NuGet. The browser's address bar shows the URL `https://docs.microsoft.com/en-us/nuget/what-is-nuget`. A cookie notice is visible at the top. The Microsoft logo and navigation links (Docs, Documentation, Learn, Code Samples, Shows, Events) are in the header. The left sidebar contains a search filter and a list of topics under 'Get started', including 'What is NuGet?'. The main content area features the article title 'An introduction to NuGet', its publication date (02/02/2022), reading time (10 minutes), and a list of user avatars. The article text describes NuGet as a tool for creating, sharing, and consuming code packages in .NET, explaining how packages are structured as ZIP files with manifests and how they are used in projects.

← → ↻ 🔒 `https://docs.microsoft.com/en-us/nuget/what-is-nuget` ⓘ ☆

ⓘ We use cookies to improve your experience on our websites and for advertising. [Privacy Statement](#)

Microsoft | Docs Documentation Learn Code Samples Shows Events

Docs / NuGet

🔍 Filter by title

What is NuGet?

▼ Get started

- Install NuGet client tools
- Install and use a package (dotnet CLI)
- Install and use a package (Visual Studio)
- Install and use a package (Visual Studio for Mac)
- Create and publish a NET Standard package (dotnet CLI)
- Create and publish a NET Standard package (Visual Studio)
- Create and publish a NET Framework package (Visual Studio)

> Consume packages

> Create packages

> Publish packages

> Concepts

> Reference

> Extensibility

> Resources

NuGet.org

An introduction to NuGet

Article • 02/02/2022 • 10 minutes to read • +6

Is this page helpful?

An essential tool for any modern development platform is a mechanism through which developers can create, share, and consume useful code. Often such code is bundled into "packages" that contain compiled code (as DLLs) along with other content needed in the projects that consume these packages.

For .NET (including .NET Core), the Microsoft-supported mechanism for sharing code is **NuGet**, which defines how packages for .NET are created, hosted, and consumed, and **provides the tools** for each of those roles.

Put simply, a NuGet package is a single ZIP file with the `.nupkg` extension that contains compiled code (DLLs), other files related to that code, and a descriptive manifest that includes information like the package's version number. Developers with code to share create packages and publish them to a public or private host. Package consumers obtain those packages from suitable hosts, add them to their projects, and then call a package's functionality in their project code. NuGet itself then handles all of the intermediate details.

Because NuGet supports private hosts alongside the public `nuget.org` host, you can use NuGet packages to share code that's exclusive to an organization or a work group. You can also use NuGet packages as a convenient way to factor your own code for use in nothing but your own projects. In short, a NuGet package is a shareable unit of code, but does not require nor imply any particular means of sharing.

qualificar

TAL

X

</>

A <PROGRAMAÇÃO