

# Início Rápido: Criar e publicar um pacote com a CLI do dotnet

Artigo • 22/12/2023

Este início rápido mostra como criar rapidamente um pacote NuGet a partir de uma biblioteca de classes .NET e publicá-lo em nuget.org usando a interface de linha de comando .NET ou [CLI do dotnet](#).

## Pré-requisitos

- O [SDK do .NET](#) , que fornece a ferramenta de linha de comando do dotnet. No Visual Studio 2017 em diante, a CLI do dotnet é instalada automaticamente com qualquer carga de trabalho relacionada ao .NET ou .NET Core.
- Uma conta gratuita em nuget.org. Siga as instruções em [Adicionar uma nova conta individual](#).

## Criar um projeto de biblioteca de classes

Você pode usar um projeto existente da Biblioteca de Classes .NET para o código que deseja empacotar, ou criar um projeto simples da seguinte maneira:

1. Crie uma pasta chamada *AppLogger*.
2. Abra um prompt de comando e vá para a pasta *AppLogger*. Todos os comandos da CLI do dotnet neste início rápido são executados na pasta atual por padrão.
3. Insira `dotnet new classlib`, o que cria um projeto com o nome da pasta atual.

Para obter mais informações, confira [dotnet new](#).

## Adicionar metadados de pacote ao arquivo de projeto

Todos os pacotes NuGet têm um manifesto que descreve seu conteúdo e suas dependências. No pacote final, o manifesto é um arquivo *.nuspec* que usa as propriedades dos metadados de NuGet que você incluiu no arquivo de projeto.

Abra o arquivo de projeto `.csproj`, `.fproj` ou `.vbproj` e adicione as propriedades a seguir dentro da tag `<PropertyGroup>` existente. Use seus próprios valores para nome e empresa e substitua o identificador de pacote por um valor exclusivo.

XML

```
<PackageId>Contoso.08.28.22.001.Test</PackageId>
<Version>1.0.0</Version>
<Authors>your_name</Authors>
<Company>your_company</Company>
```

### 📘 Importante

O identificador de pacote deve ser exclusivo em [nuget.org](https://nuget.org) e em outras origens de pacotes. A publicação torna o pacote publicamente visível. Portanto, se você usar a biblioteca AppLogger de exemplo ou outra biblioteca de teste, use um nome exclusivo que inclua `Sample` ou `Test`.

É possível adicionar propriedades opcionais descritas em [Propriedades de metadados do NuGet](#).

### ⚠️ Observação

Para pacotes que você cria para consumo público, preste atenção especial à propriedade `PackageTags`. As tags ajudam outras pessoas a encontrar seu pacote e entender o que ele faz.

## Executar o comando pack

Para compilar um pacote do NuGet ou arquivo `.nupkg` do projeto, execute o comando [dotnet pack](#), que também compila o projeto automaticamente.

CLI do .NET

```
dotnet pack
```

A saída mostra o caminho até o arquivo `.nupkg`.

Saída

```
MSBuild version 17.3.0+92e077650 for .NET
Determining projects to restore...
Restored C:\Users\myname\source\repos\AppLogger\AppLogger.csproj (in 64 ms).
AppLogger ->
C:\Users\myname\source\repos\AppLogger\bin\Debug\net6.0\AppLogger.dll
Successfully created package
'C:\Users\myname\source\repos\AppLogger\bin\Debug\Contoso.08.28.22.001.Test.1.0
.0.nupkg'.
```

## Gerar pacote automaticamente no build

Para executar `dotnet pack` automaticamente sempre que você executar `dotnet build`, adicione a seguinte linha ao arquivo de projeto em `<PropertyGroup>`:

XML

```
<GeneratePackageOnBuild>true</GeneratePackageOnBuild>
```

## Publicar o pacote

Publique seu arquivo `.nupkg` em nuget.org usando o comando `dotnet nuget push` com uma chave de API obtida por você em nuget.org.

### ⓘ Observação

- Nuget.org verifica todos os pacotes carregados em busca de vírus e rejeita os pacotes quando encontra um vírus. Nuget.org também verifica periodicamente todos os pacotes listados.
- Os pacotes que você publicar em nuget.org também são publicamente visíveis para outros desenvolvedores, a menos que você os remova da lista. Para hospedar pacotes de forma privada, consulte [Hospedar seus próprios feeds NuGet](#).

## Obter sua chave de API

1. [Entre em sua conta de nuget.org](#) ou [crie uma conta](#) caso ainda não tenha uma.
2. Selecione seu nome de usuário no canto superior direito e selecione **Chaves de API**.
3. Selecione **Criar** e forneça um nome para sua chave.
4. Em **Selecionar Escopos**, selecione **Push**.
5. Em **Selecionar Pacotes** > **Padrão Glob**, insira **\***.
6. Selecione **Criar**.
7. Selecione **Copiar** para copiar a nova chave.

⚠ Your API key has been regenerated. Make sure to copy your new API key now using the **Copy** button below. You will not be able to do so again.



test-key1

🕒 Expires in a year

🔗 Push new packages and package versions

**Package owner:** NuGet-test

**Glob pattern:** \*



Copy



Edit



Regenerate



Delete

### 📘 Importante

- Sempre mantenha sua chave de API em segredo. A chave de API é como uma senha que permite que qualquer pessoa gerencie pacotes em seu nome. Exclua ou gere novamente sua chave de API se ela for revelada acidentalmente.
- Salve sua chave em um local seguro, pois não será possível copiá-la novamente no futuro. Se você retornar à página da chave de API, será necessário gerar novamente a chave para copiá-la. Também é possível remover a chave de API, se você não quiser mais fazer o push de pacotes.

Os *Escopos* permitem criar chaves de API separadas para finalidades diferentes. Cada chave tem um período de expiração e pode ter o escopo definido para pacotes ou padrões glob específicos. Você também define o escopo de cada chave para operações específicas:

enviar novos pacotes e versões de pacotes, enviar por push apenas novas versões de pacotes ou remover da lista.

Por meio de escopo, é possível criar chaves de API para diferentes pessoas que gerenciam os pacotes para a sua organização, de modo que elas tenham somente as permissões necessárias.

Para saber mais, confira [Chaves de API com escopo](#).

## Publicar com dotnet nuget push

Na pasta que contém o *arquivo .nupkg*, execute o comando a seguir. Especifique o nome do arquivo *.nupkg* e substitua o valor da chave pela chave da API.

CLI do .NET

```
dotnet nuget push Contoso.08.28.22.001.Test.1.0.0.nupkg --api-key  
qz2jga8p13dvn2akksyquwcs9ygggg4exypy3bhxy6w6x6 --source  
https://api.nuget.org/v3/index.json
```

A janela de saída mostra os resultados do processo de publicação.

Saída

```
Pushing Contoso.08.28.22.001.Test.1.0.0.nupkg to  
'https://www.nuget.org/api/v2/package'...  
PUT https://www.nuget.org/api/v2/package/  
warn : All published packages should have license information specified. Learn  
more: https://aka.ms/nuget/authoring-best-practices#licensing.  
Created https://www.nuget.org/api/v2/package/ 1221ms  
Your package was pushed.
```

Para obter mais informações, confira [dotnet nuget push](#).

### ⚠ Observação

Para evitar que seu pacote de teste fique permaneça em nuget.org, você pode enviá-lo por push para o site de teste de nuget.org em <https://int.nugettest.org>. Observe que os pacotes carregados para int.nugettest.org podem não ser preservados.

## Erros de publicação

Os erros do comando `push` geralmente indicam o problema. Por exemplo, talvez você tenha esquecido de atualizar o número de versão em seu projeto e, portanto, está tentando publicar um pacote que já existe.

Você também verá erros se sua chave de API for inválida ou tiver expirado, ou se você tentar publicar um pacote usando um identificador que já existe no host. Suponha, por exemplo, que o identificador `AppLogger-test` já exista em [nuget.org](https://nuget.org). Se você tentar publicar um pacote com esse identificador, o comando `push` resultará no seguinte erro:

### Saída

```
Response status code does not indicate success: 403 (The specified API key is invalid, has expired, or does not have permission to access the specified package.).
```

Se você receber esse erro, verifique se está usando uma chave de API válida que ainda não expirou. Se estiver, o erro indicará que o identificador do pacote já existe no host. Para corrigir o erro, altere o identificador do pacote para torná-lo único, recompile o projeto, recrie o arquivo `.nupkg` e tente novamente o comando `push`.

## Gerenciar o pacote publicado

Quando seu pacote for publicado com êxito, você receberá um e-mail de confirmação. Para ver o pacote que você acabou de publicar, em [nuget.org](https://nuget.org), selecione seu nome de usuário no canto superior direito e selecione **Gerenciar pacotes**.

### ⚠ Observação



Poderá demorar algum tempo para o pacote ser indexado e aparecer nos resultados da pesquisa, onde outras pessoas podem encontrá-lo. Durante esse tempo, o pacote aparece em **Pacotes não listados** e a página do pacote mostra a seguinte mensagem:

**⚠ This package has not been published yet.** It will appear in search results and will be available for install/restore after both validation and indexing are complete. Package validation and indexing may take up to an hour. [Read more.](#)

Você acabou de publicar um pacote NuGet em nuget.org, o qual pode ser usado por outros desenvolvedores podem usar em seus próprios projetos.

Se você criou um pacote que não é útil (como este pacote de exemplo que foi criado com uma biblioteca de classes vazia) ou decidiu que não deseja que o pacote fique visível, você poderá *remover o pacote da lista* para ocultá-lo dos resultados da pesquisa:

1. Depois que o pacote aparecer em **Pacotes Publicados** na página **Gerenciar Pacotes**, selecione o ícone de lápis ao lado da listagem de pacotes.

Published Packages					
1 package / 0 downloads					
Package ID	Owners	Signing Owner	Downloads	Latest Version	
 Contoso.08.28.22.001.Test	Test	username (0 certificates)	0	1.0.0	

2. Na próxima página, selecione **Listagem**, desmarque a caixa de seleção **Listar nos resultados da pesquisa** e selecione **Salvar**.

Listing

Select version

1.0.0 (Latest) ▼

List or unlist version

You can control how your packages are listed using the checkbox below. As per policy, permanent deletion is not supported as it would break every project depending on the availability of the package. For more assistance, [Contact Support](#).

☐ List in search results.

Unlisted packages cannot be installed directly and do not show up in search results.

Save

O pacote agora aparece em **Pacotes não listados** em **Gerenciar pacotes** e não aparece mais nos resultados da pesquisa.

### ⓘ Observação

Para evitar que seu pacote de teste fique ativo em nuget.org, você pode enviá-lo por push para o site de teste de nuget.org em <https://int.nugettest.org> . Observe que os pacotes carregados para int.nugettest.org podem não ser preservados.

Parabéns por criar e publicar seu primeiro pacote NuGet!

## Vídeo relacionado

<https://learn.microsoft.com/shows/NuGet-101/Create-and-Publish-a-NuGet-Package-with-the-NET-CLI-5-of-5/player>

Encontre mais vídeos sobre o NuGet no [Channel 9](#) e no [YouTube](#) .

## Próximas etapas

Veja mais detalhes sobre como criar pacotes com a CLI do dotnet:

[Criar um pacote NuGet com a CLI do dotnet](#)

Obtenha mais informações sobre como criar e publicar pacotes NuGet:

- [Publicar um pacote](#)
- [Pacotes de pré-lançamento](#)
- [Suporte a várias estruturas de destino](#)
- [Controle de versão do pacote](#)
- [Adicionar uma expressão ou arquivo de licença](#)
- [Criar pacotes localizados](#)
- [Criar pacotes de símbolos](#)
- [Assinar pacotes](#)

## Comentários

Esta página foi útil?

 Yes

 No



[Fornecer comentários sobre o produto](#)