Package and Publish To NuGet

Create a NuGet package with the dotnet CLI

Full article: https://learn.microsoft.com/en-us/nuget/create-packages/creating-a-package-dotnet-cli

Set Properties

Set (at least) the following properties in the .csproj file:

```xml
<Project Sdk="Microsoft.NET.Sdk">
  <PropertyGroup>
    <TargetFramework>netstandard2.0</TargetFramework>
    <PackageId>UniqueID</PackageId>
    <Version>1.0.0</Version>
    <Authors>Author Name</Authors>
    <Company>Company Name</Company>
    <Product>Product Name</Product>
  </PropertyGroup>
</Project>
```

PackageId will be used as the identifier on nuget.org. For example, if you give PackageId the value MyFancyLib, then the published nuget will be at https://www.nuget.org/packages/MyFancyLib.

PackageId must be unique, and not used by anyone else previously. Moreover, each time you publish an update the PackageId / Version combination must be unique. (You cannot overwrite a previously published version.)

Run the pack command

To build the NuGet package or .nupkg file, run the `dotnet pack` command from the project folder, which also builds the project automatically.

```
dotnet pack
```

The output shows the path to the .nupkg file:

```
MSBuild version 17.3.0+92e077650 for .NET
  Determining projects to restore...
  Restored C:\projects\MyFancyLib\MyFancyLib.csproj (in 97 ms).
  Successfully created package
'C:\MyFancyLib\bin\Debug\MyFancyLib.1.0.0.nupkg'.
```

Publish NuGet packages

Full article: https://learn.microsoft.com/en-us/nuget/nuget-org/publish-a-package

To publish on nuget.org, sign in to nuget.org with a Microsoft account, and use it to create a free nuget.org account.

Create an API key

1. Sign into your nuget.org account or create an account if you don't have one already.
2. Select your user name at upper right, and then select API Keys.
3. Select Create, and provide a name for your key.
4. Under Select Scopes, select Push.
5. Under Select Packages > Glob Pattern, enter *.
6. Select Create.
7. Select Copy to copy the new key.

Use the dotnet CLI to publish

From the folder that contains the .nupkg file, run the following command. Specify your .nupkg filename, and replace the key value with your API key.

```
dotnet nuget push MyFancyLib.1.0.0.nupkg --api-key
qz2jga8pl3dvn2akksyquwcs9ygggg4exypy3bhxy6w6x6 --source
https://api.nuget.org/v3/index.json
```

The output shows the results of the publishing process:

```
Pushing MyFancyLib.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.
```

dotnet