NuGet Packages

## Nuget.config

This file needs to be put in the same folder as the solution file, usually in the root of the repository.

<?xml version="1.0" encoding="utf-8"?>

<configuration>

<packageSources>

<clear />

<add key="nuget" value="https://api.nuget.org/v3/index.json" />

</packageSources>

</configuration>

## .csproj

In the .csproj file, add the following between the PropertyGroup tags:

<RepositoryUrl>https://github.com/S65-2-project/MessageBroker</RepositoryUrl>

<Version>1.0.1</Version>

<IncludeSymbols>false</IncludeSymbols>

## dotnet pack

Pack the Release build (-c/--configuration) into the release output folder (-o/--output

dotnet pack -c Release -o release

## dotnet nuget push

Deliver NuGet package to nuget.org

dotnet nuget push ./release/\*.nupkg --source "nuget" --api-key ${{ secrets.NUGET\_API\_KEY }} --skip-duplicate

### --skip-duplicate

When pushing multiple packages to an HTTP(S) server, treats any 409 Conflict response as a warning so that the push can continue. Available since .NET Core 3.1 SDK.

## Symbol Packages

A good debugging experience relies on the presence of debug symbols as they provide critical information like the association between the compiled and the source code, names of local variables, stack traces, and more. You can use symbol packages (.snupkg) to distribute these symbols and improve the debugging experience of your NuGet packages.

If you want a symbol package (snupkg) as well, use the following settings:

<IncludeSymbols>true</IncludeSymbols>

<SymbolPackageFormat>snupkg</SymbolPackageFormat>