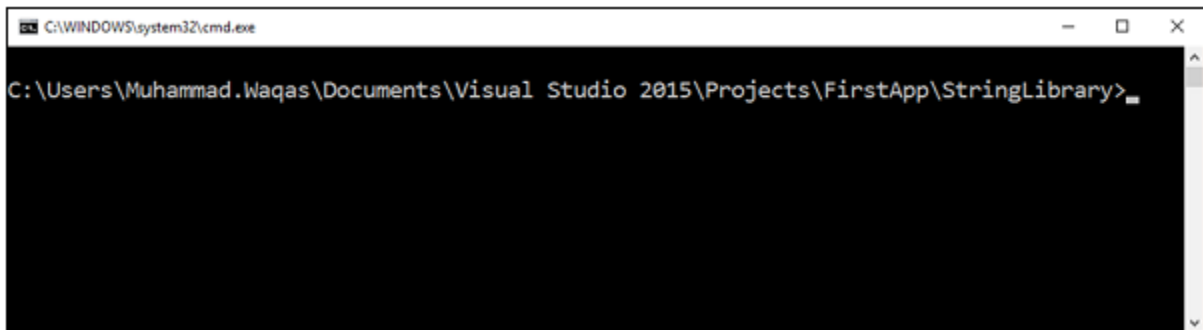


به اشتراک گذاشتن کتابخانه ها در Net Core.

در این بخش به چگونگی اشتراک گذاری کتابخانه ی خود به صورت NuGet Package می پردازیم؛ به گونه ای که بتوان از این کتابخانه در پروژه های دیگر نیز استفاده کرد. ایجاد یک پکیج از کدی که می خواهید بسته بندی کنید و با دیگر بخش ها به اشتراک بگذارید و یا از طریق گالری همگانی nuget.org یا از طریق یک گالری خصوصی داخل تشکیلات شما آغاز می شود. این پکیج می تواند فایل های بیشتری همچون `readme` را نیز شامل شود. این فایل زمانی نمایش داده می شود که پکیج نصب شده باشد و می تواند برای فایل های پروژه ای مشخصی کار تبدیل را انجام دهد.

حالا مثال ساده ای را در نظر بگیرید که در آن خواهیم از کتابخانه ی خود یک بسته ی NuGet را ایجاد کنیم. برای انجام این کار `cmd` را باز کنید و به پوشه ای بروید که فایل `project.json` پروژه ی کتابخانه ی شما در آن واقع است.



```
C:\WINDOWS\system32\cmd.exe
C:\Users\Muhammad.Waqas\Documents\Visual Studio 2015\Projects\FirstApp\StringLibrary>
```

حالا دستور زیر را وارد کنید.

```
dotnet help
```

```
C:\WINDOWS\system32\cmd.exe
C:\Users\Muhammad.Waqas\Documents\Visual Studio 2015\Projects\FirstApp\StringLibrary>dotnet help

.NET Command Line Tools (1.0.0-preview2-003131)
Usage: dotnet [host-options] [command] [arguments] [common-options]

Arguments:
  [command]           The command to execute
  [arguments]         Arguments to pass to the command
  [host-options]      Options specific to dotnet (host)
  [common-options]    Options common to all commands

Common options:
  -v|--verbose        Enable verbose output
  -h|--help           Show help

Host options (passed before the command):
  -v|--verbose        Enable verbose output
  --version           Display .NET CLI Version Number
  --info             Display .NET CLI Info

Common Commands:
  new                Initialize a basic .NET project
  restore            Restore dependencies specified in the .NET project
  build              Builds a .NET project
  publish            Publishes a .NET project for deployment (including the runtime)
  run                Compiles and immediately executes a .NET project
  test              Runs unit tests using the test runner specified in the project
  pack              Creates a NuGet package

C:\Users\Muhammad.Waqas\Documents\Visual Studio 2015\Projects\FirstApp\StringLibrary>_
```

در نهایت می توانید دستورات مختلفی همچون `new`، `restore`، `build` و ... را مشاهده کنید.

آخرین دستور `pack` است. کار این دستور ایجاد یک بسته ی NuGet است. حالا دستور زیر را وارد کنید.

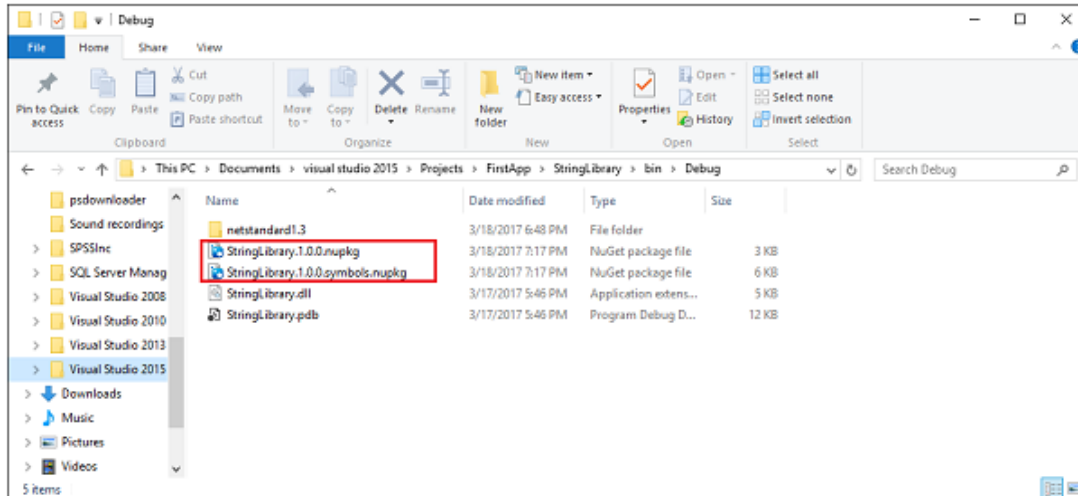
```
dotnet pack
```

```
C:\WINDOWS\system32\cmd.exe
C:\Users\Muhammad.Waqas\Documents\Visual Studio 2015\Projects\FirstApp\StringLibrary>dotnet pack

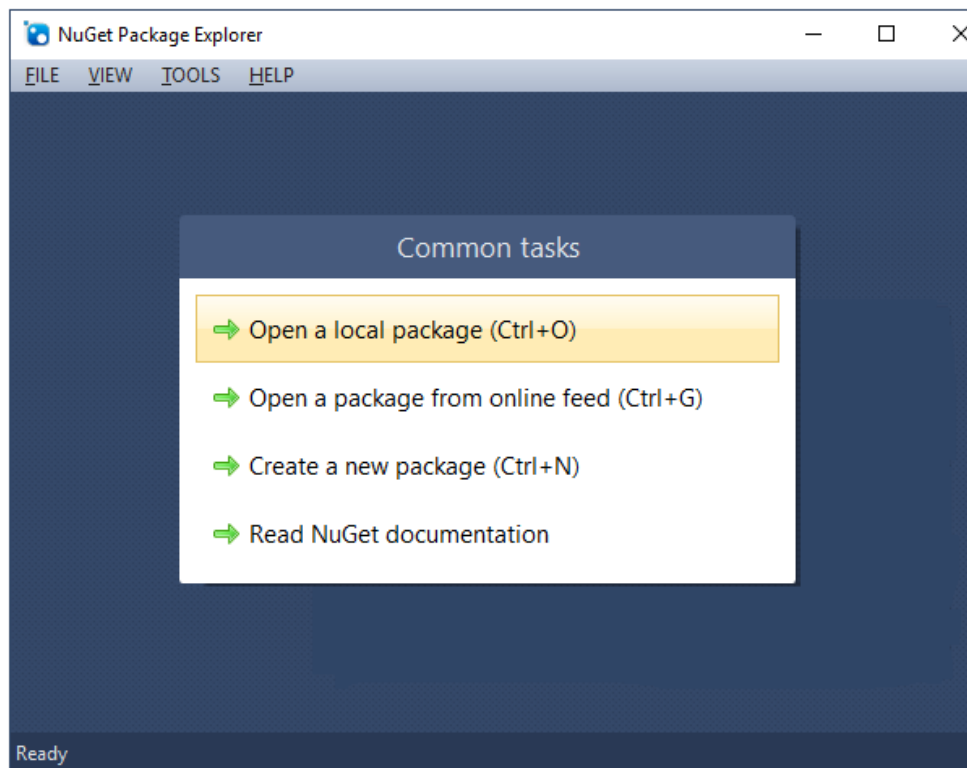
Project StringLibrary (.NETStandard,Version=v1.3) was previously compiled. Skipping compilation.
Producing nuget package "StringLibrary.1.0.0" for StringLibrary
StringLibrary -> C:\Users\Muhammad.Waqas\Documents\Visual Studio 2015\Projects\FirstApp\StringLi
brary\bin\Debug\StringLibrary.1.0.0.nupkg
Producing nuget package "StringLibrary.1.0.0.symbols" for StringLibrary
StringLibrary -> C:\Users\Muhammad.Waqas\Documents\Visual Studio 2015\Projects\FirstApp\StringLi
brary\bin\Debug\StringLibrary.1.0.0.symbols.nupkg

C:\Users\Muhammad.Waqas\Documents\Visual Studio 2015\Projects\FirstApp\StringLibrary>_
```

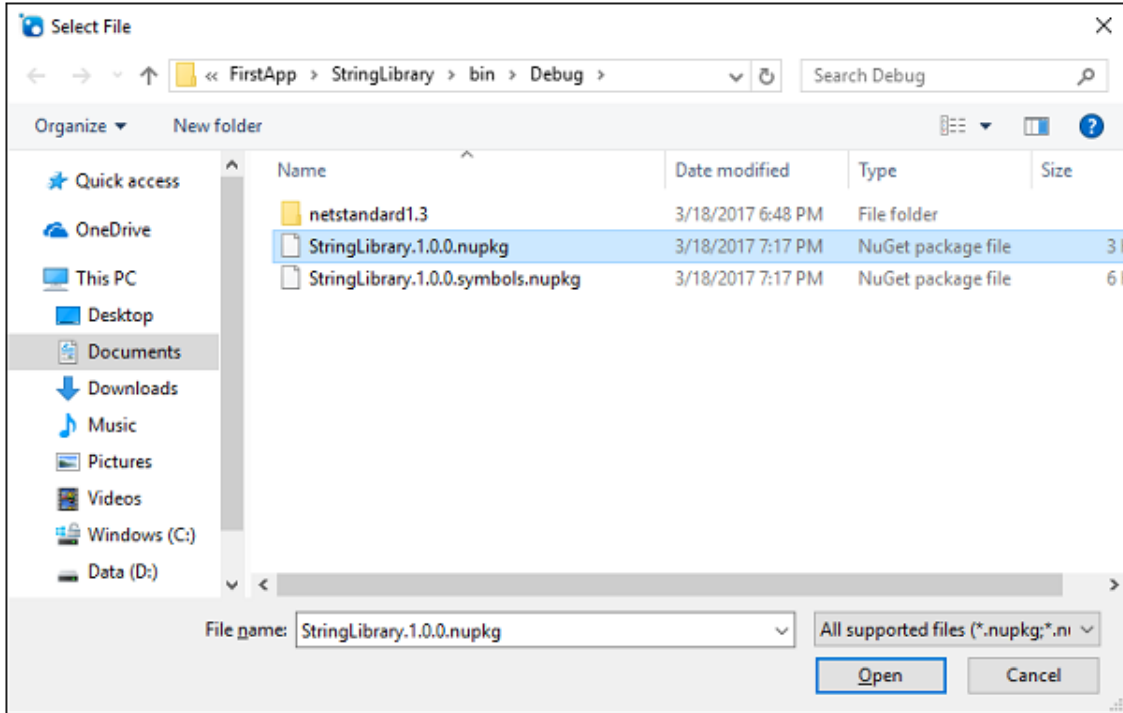
همان طور که می بینید بسته های NuGet در پوشه ی `bin` ایجاد شده اند. پوشه ی `bin\Debug` را باز کنید.



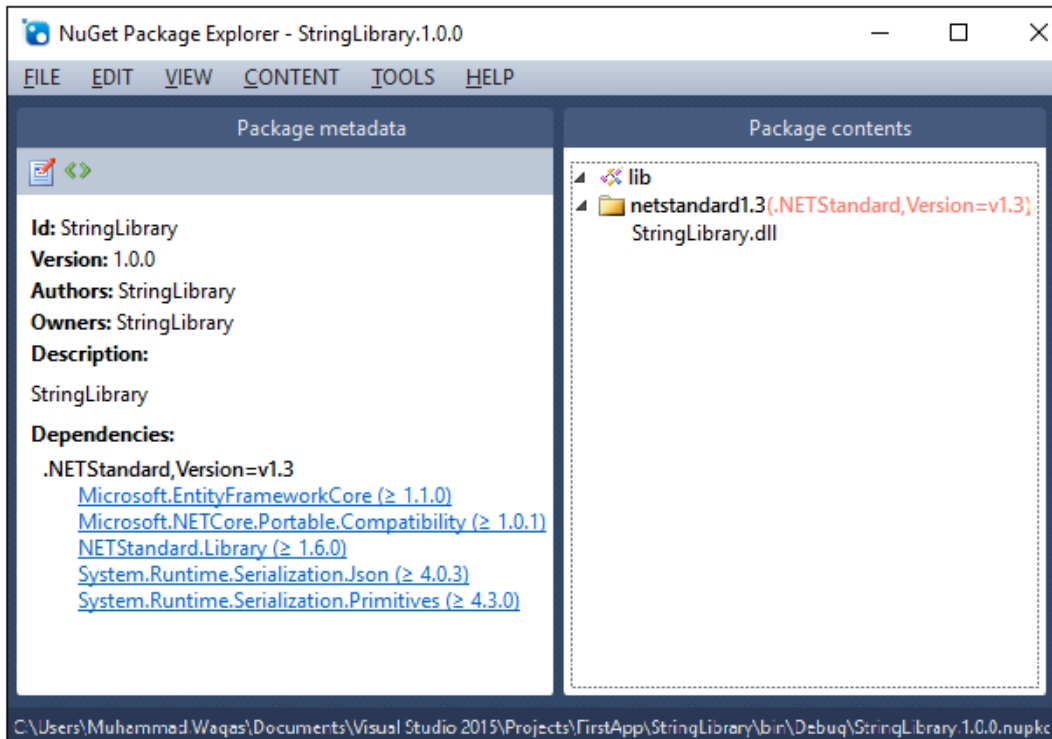
حالا سؤال اینجاست که چه چیزی داخل بسته های NuGet است؟ برای پی بردن به این موضوع می توانیم از NuGet Package Explorer استفاده کنیم؛ آن را باز کنید.



اولین گزینه ی Open a local package را انتخاب کنید.

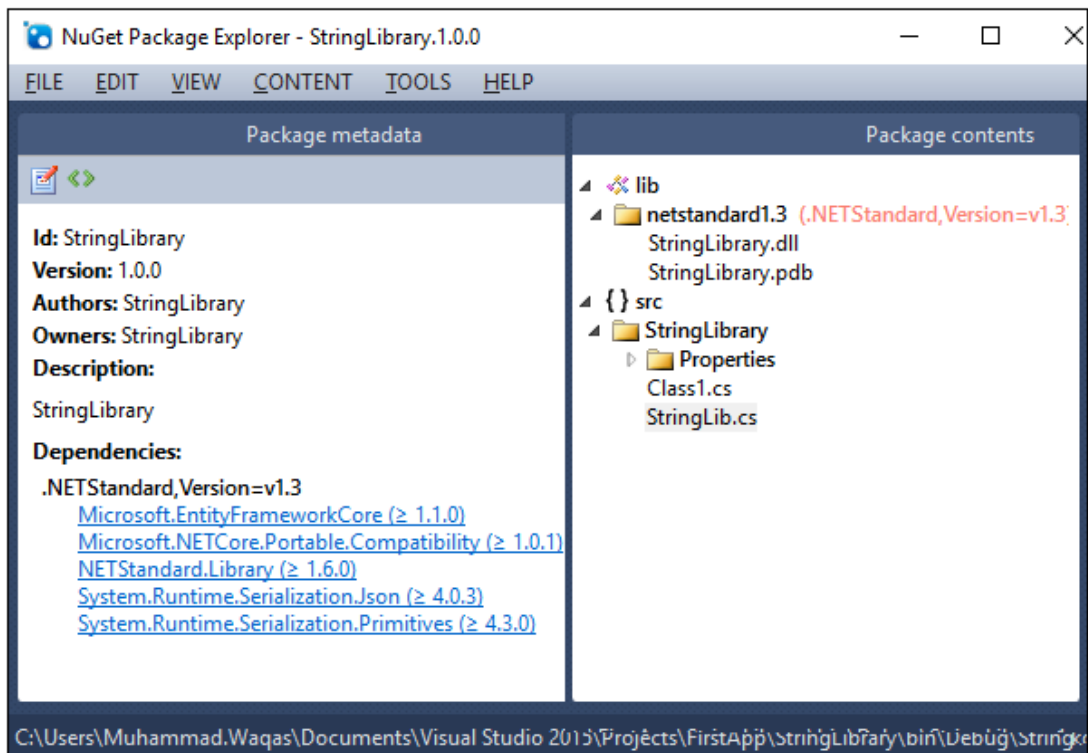


StringLibrary.1.0.0.nupkg را انتخاب کنید و بر روی Open کلیک کنید.

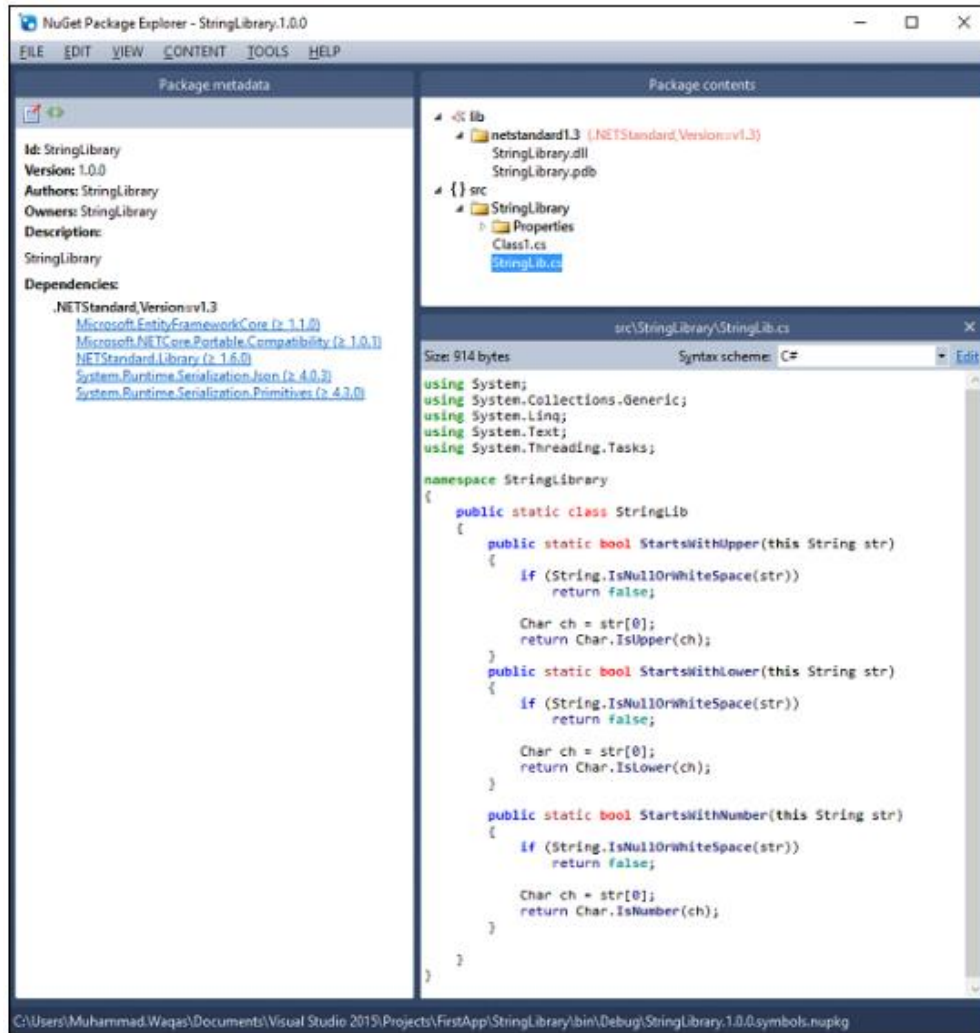


همان طور که می بینید در بخش Package contents تنها StringLibrary.dll وجود دارد. در بخش Package metadata اطلاعات کمی در رابطه با این کتابخانه وجود دارد، مانند Id و نسخه های تمامی وابستگی ها.

حالا StringLibrary.1.0.0.symbols.nupkg را باز کنید.



در این بسته ی NuGet می توانید فایل های منبع و فایل pdb* را نیز ببینید. اگر بر روی فایل StringLib.cs دوبار کلیک کنید، در این صورت می توانید سورس کد را هم ببینید.



در اینجا سوآلی که مطرح می شود این است که چگونه می توان متادیتاهایی همچون version ،authors ،description و ... را پیکربندی کرد.

برای تعریف متادیتاهای پروژه اطلاعات کامپایل و وابستگی ها در پروژه های NET Core . از فایل project.json استفاده می شود. فایل project.json را باز کنید و اطلاعات زیر را به آن اضافه کنید.

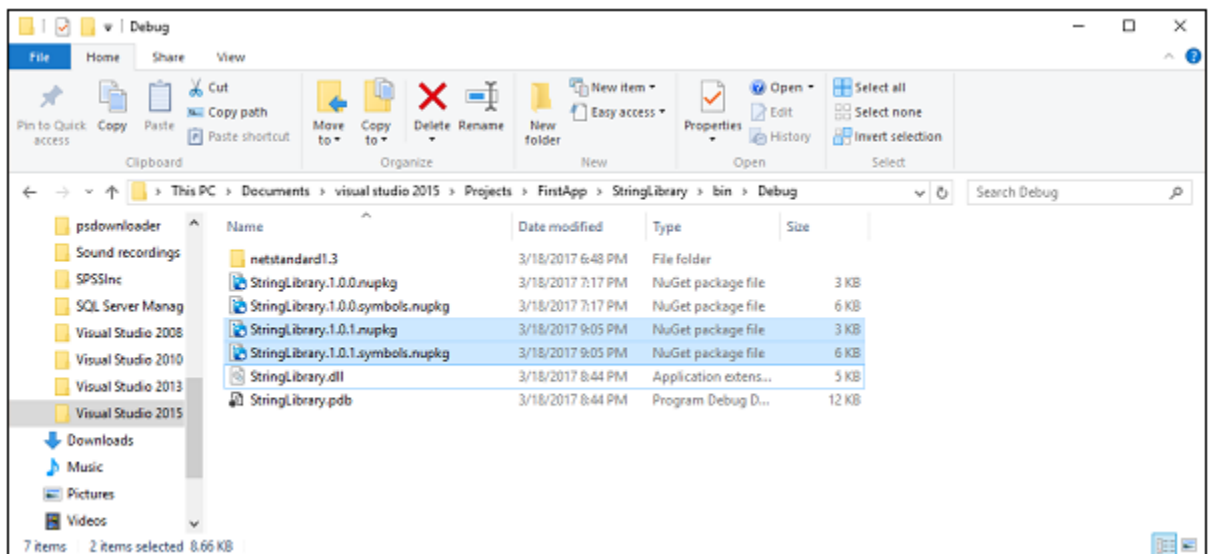
```
{
  "authors": [ "Mark Junior" ],
  "description": "String Library API",
  "version": "1.0.1-*",
  "supports": {},
```

```

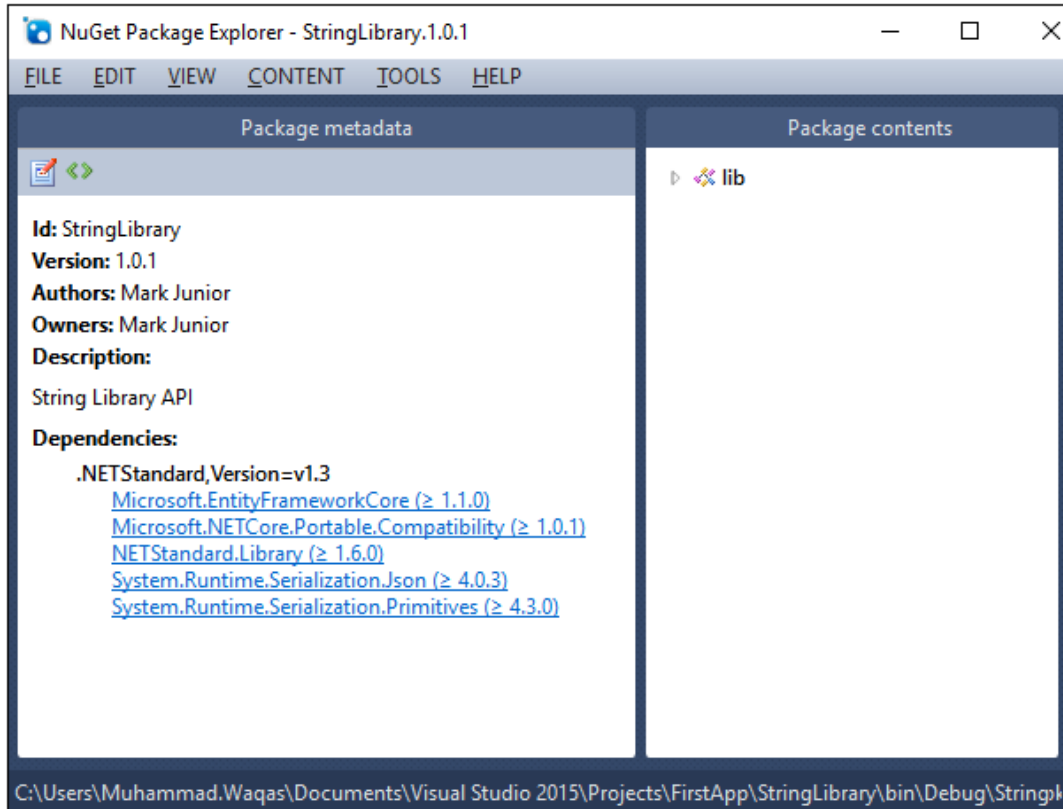
"dependencies": {
  "Microsoft.EntityFrameworkCore": "1.1.0",
  "Microsoft.NETCore.Portable.Compatibility": "1.0.1",
  "NETStandard.Library": "1.6.0",
  "System.Runtime.Serialization.Json": "4.0.3",
  "System.Runtime.Serialization.Primitives": "4.3.0"
},
"frameworks": {
  "netstandard1.3": {}
}
}

```

حالا می توانید این اطلاعات را در اینجا مشاهده کنید. این فایل را ذخیره کنید، پروژه ی کتابخانه را بسازید و برای بار چندم دستور “dotnet pack” را اجرا کنید.



داخل پوشه ی bin\Debug می توانید بسته های نوگت StringLibrary را ببینید که با نسخه ی 1.0.1 ایجاد شده اند. آن را در NuGet Package Explorer باز کنید.



حالا می توانید متادیتای به روز شده را مشاهده کنید. سؤال این است که چگونه می توان در بسته ی دیگری از آن استفاده کرد؟

حالا باید در جایی از فید نوگت از publish استفاده کنیم و بعد از آن می توانیم در پروژه ی دیگری از آن استفاده کنیم.

دو گزینه وجود دارد که می توان به کمک آن ها متادیتای به روز شده را منتشر کرد:

- انتشار آن در [nuget.org](https://www.nuget.org)
- جای دادن متادیتا در فید خصوصی نوگت

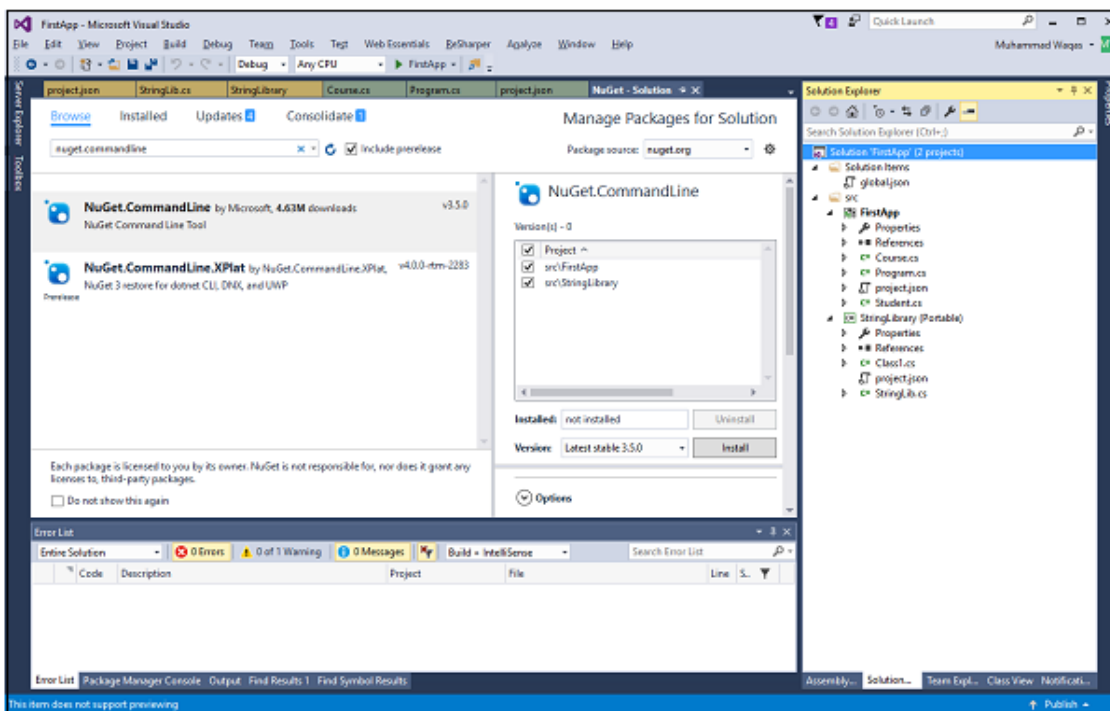
در اینجا ما از گزینه ی دوم استفاده می کنیم، زیرا ساختن اکانت در [nuget.org](https://www.nuget.org) دشوارتر از این گزینه است. برای آن که به چگونگی انتشار بسته ی خود در [nuget.org](https://www.nuget.org) پی ببرید، می توانید از دستورات عمل های لینک زیر استفاده کنید:

<https://docs.microsoft.com/en-us/nuget/create-packages/publish-a-package>

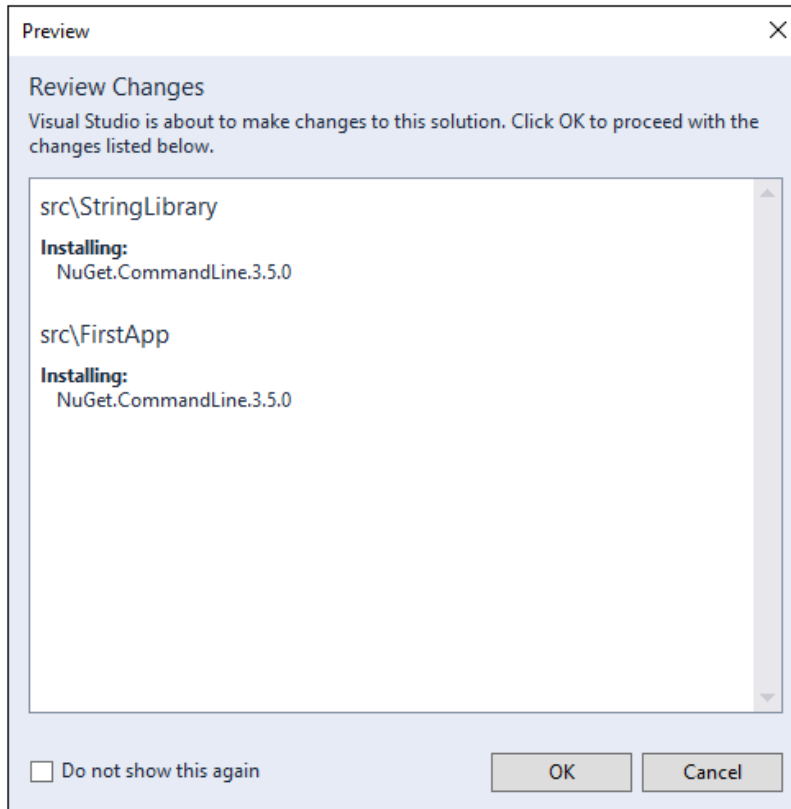
جهت جای دادن متادیتای به روز شده در فید خصوصی نوگت مراحل زیر را دنبال کنید:

مرحله 1: برای شروع به ابزار nuget commandline نیاز داریم و باید آن را نصب کنیم. NuGet Package Manager را باز کنید و nuget.commandline را جستجو کنید.

مرحله 2: NuGet.Commandline را انتخاب کنید و بر روی Install کلیک کنید.



مرحله 3: بر روی OK کلیک کنید تا نصب NuGet.Commandline آغاز شود. همچنین برای نصب دستی آن می توانید آن را از آدرس <https://dist.nuget.org/index.html> دانلود کنید و پس از آن متغیر محیط را برپا کنید.



مرحله 4: بعد از تمام شدن نصب، برای بار دوم `cmd` را باز کنید و به پوشه `bin\Debug` که بسته های نوگت در آن قرار دارند بروید و دستور زیر را مشخص کنید.

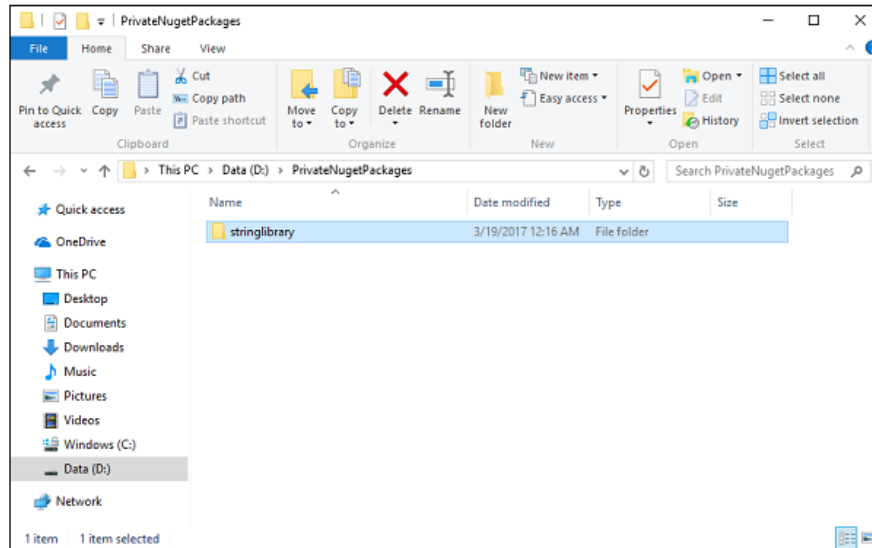
```
nuget add StringLibrary.1.0.1.nupkg -Source D:\PrivateNugetPackages
```

مرحله 5: در دستور بالا بسته `StringLibrary.1.0.1.nupkg` را به فید خصوصی خود اضافه می کنیم که مکان آن `D:\PrivateNugetPackages` و منبع بسته را مشخص می کند.

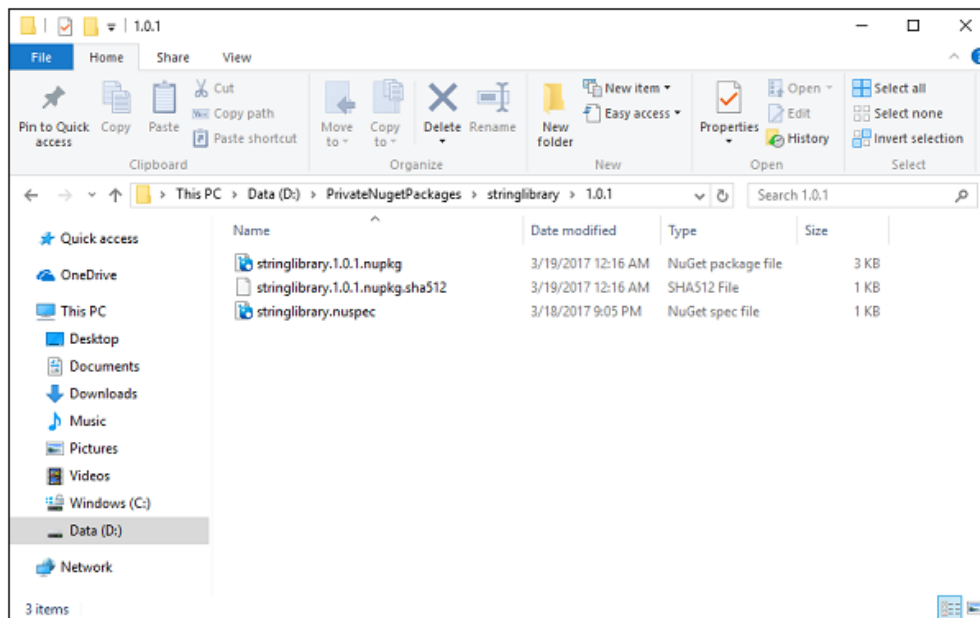
مرحله 6: همان طور که می بینید `StringLibrary` نصب شده است و می توان آن را به فید خصوصی خود اضافه کرد.

```
C:\WINDOWS\system32\cmd.exe
C:\Users\Muhammad.Waqas\Documents\Visual Studio 2015\Projects\FirstApp\StringLibrary\bin\Debug> nuget add StringLibrary.1.0.1.nupkg -Source D:\PrivateNugetPackages
Installing StringLibrary 1.0.1.
Successfully added package 'StringLibrary.1.0.1.nupkg' to feed 'D:\PrivateNugetPackages'.
C:\Users\Muhammad.Waqas\Documents\Visual Studio 2015\Projects\FirstApp\StringLibrary\bin\Debug>
```

مرحله 7: به آن پوشه بروید.



مرحله 8: داخل پوشه ی stringlibrary می توانید پوشه ی دیگری را همراه با اسم نسخه ببینید. همان طور که می بینید همان نسخه ی 1.0.1 در اینجا قرار دارد.



بسته ی NuGet در اینجا قرار گرفته است.