

آموزش MVC Core - اضافه کردن Class Library در MVC Core با استفاده از NET Core Command-Line Interface (CLI)

معرفی

این مقاله نحوه اضافه کردن "Class Library" به "ASP.NET Core" را با استفاده از "NET Core Command-Line Interface (CLI)" به روش ساده توضیح می دهد.

.Net Core CLI

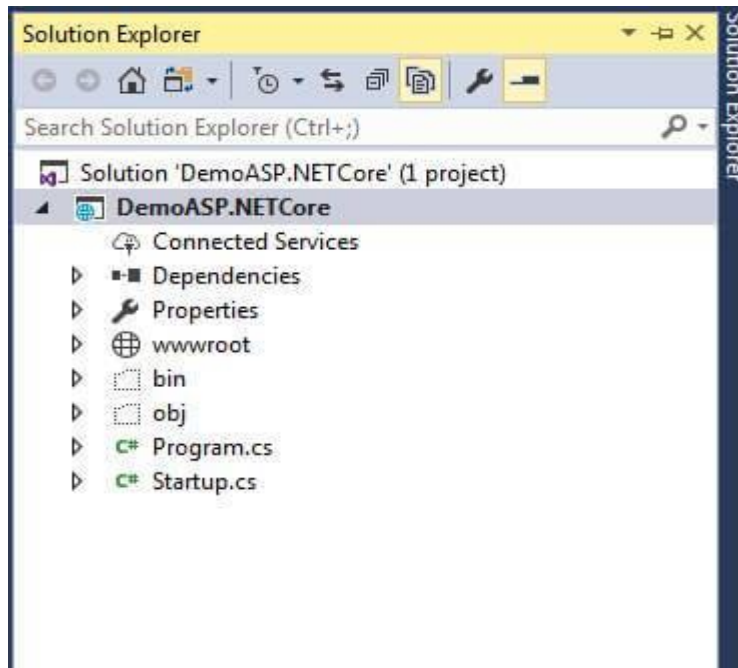
NET Core command-line interface (CLI) یک خط فرمان جدید "cross-platform" برای توسعه برنامه های NET است. با استفاده از "NET Core CLI" می توانیم برنامه جدید "NET core" باز کنیم، برنامه های کاربردی را تغییر دهیم یا "class libraries" اضافه کنیم و غیره. ما می توانیم برنامه ما را با استفاده از خط فرمان (command line) مدیریت کنیم.

زمینه

ما برنامه های ASP.NET Core را با استفاده از رابط کاربری گرافیکی (GUI) مدیریت می کنیم. هر زمان که ما یک برنامه را توسعه می دهیم، کارهای بسیاری انجام می دهیم مانند اضافه کردن فایل ها، کامپایل برنامه و اجرای برنامه با استفاده از رابط کاربری گرافیکی؛ اما ما می توانیم این کارها را با استفاده از command line انجام دهیم. با استفاده از NET Core CLI می توانیم برنامه ASP.NET Core ما را مدیریت کنیم.

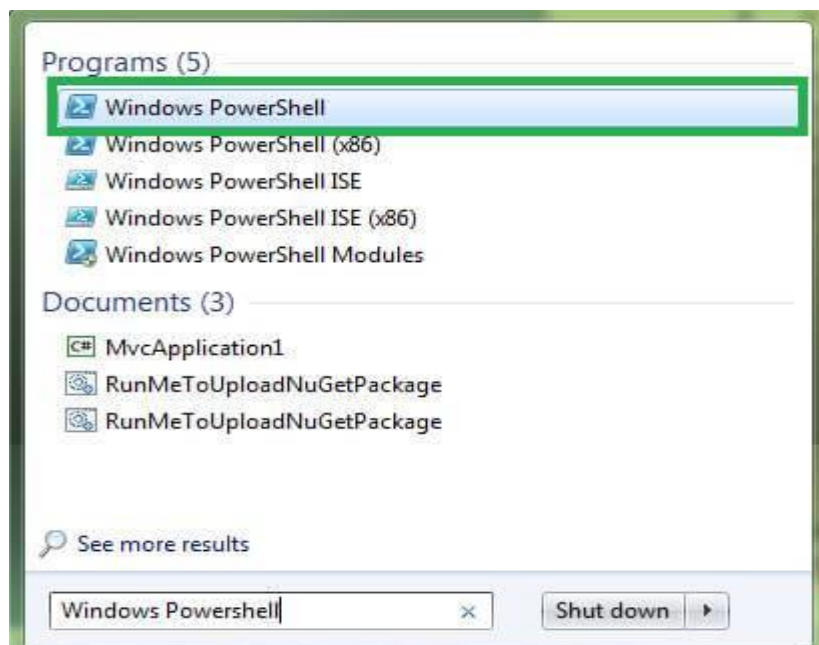
مراحل اضافه کردن Class Library با استفاده از CLI

اول، یک "ASP.NET core application" جدید باز کنید و یک صفحه نمونه اضافه کنید و سپس برنامه را بسازید و اجرا کنید. به solution explorer بروید در آنجا می توانید یک پروژه ببینید.

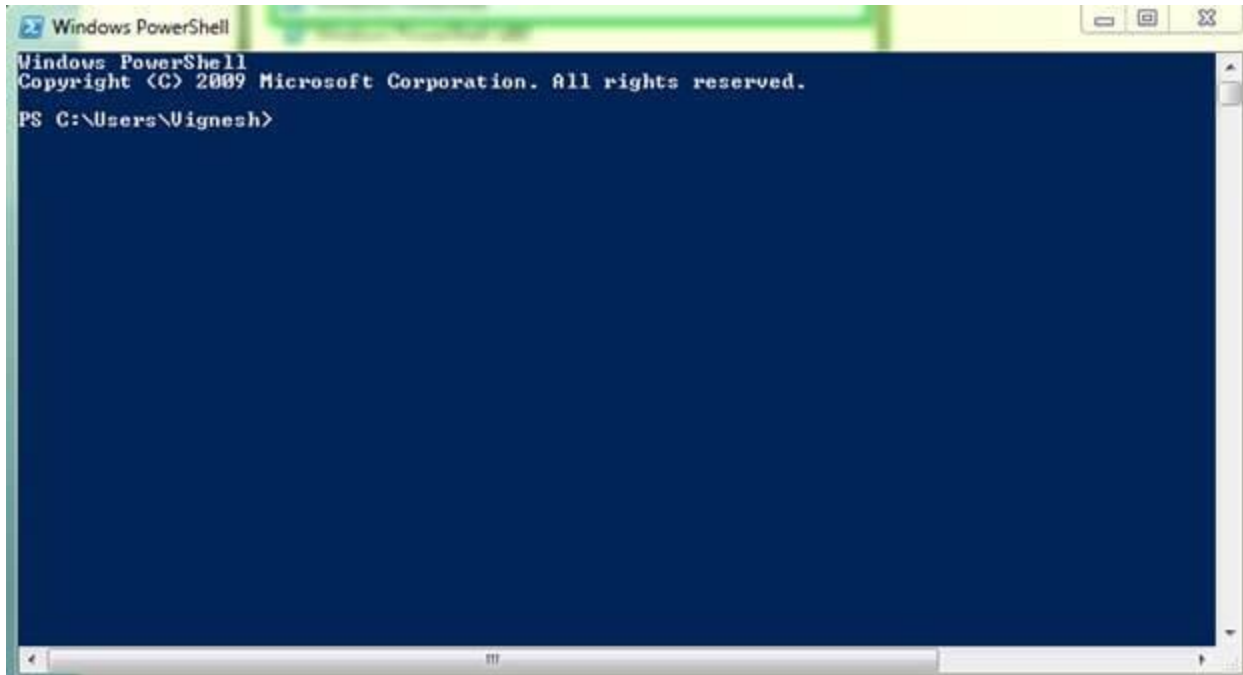


باز کردن PowerShell

حالا ما می خواهیم با استفاده از برخی از مراحل ، با کمک "NET Core CLI" ، "class library" اضافه کنیم. Windows PowerShell را باز کنید:



حالا پنجره Windows PowerShell باز خواهد شد.



```
Windows PowerShell
Copyright (C) 2009 Microsoft Corporation. All rights reserved.
PS C:\Users\Vignesh>
```

محل پروژه (Project Location)

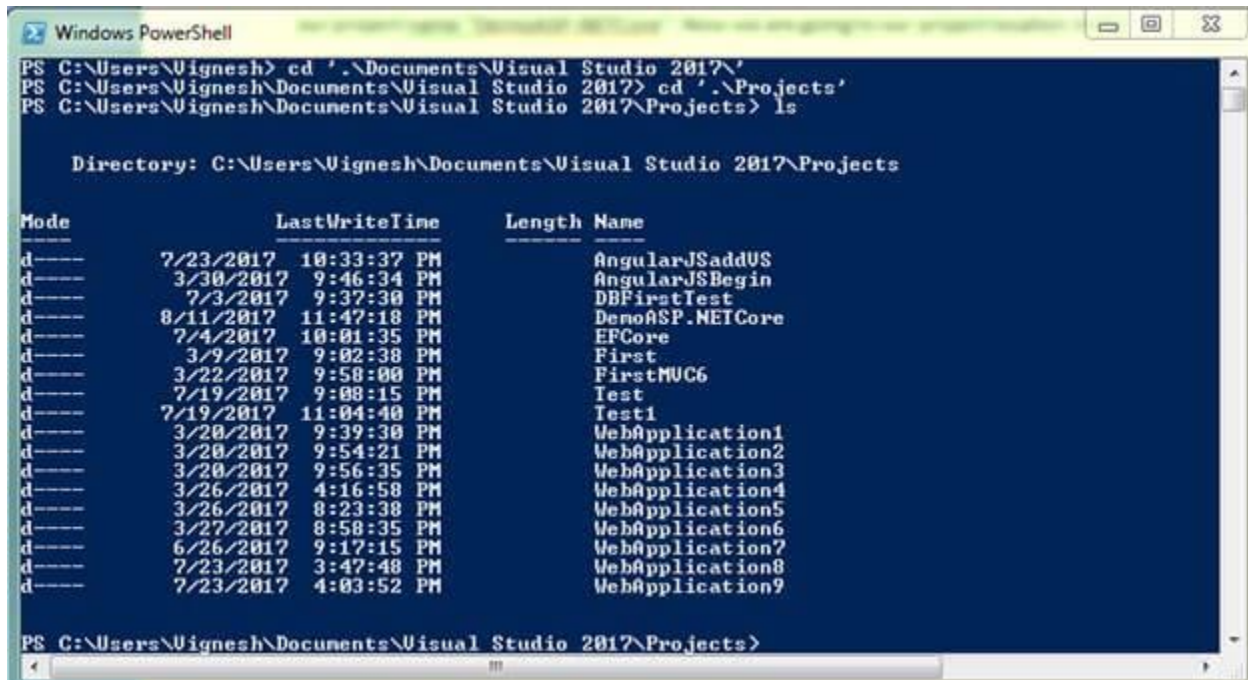
پروژه در مسیر (C:\Users\Vignesh\Documents\Visual Studio 2017\Projects) ذخیره شده و نام پروژه DemoASP.NETCore می باشد. در حال حاضر ما به محل پروژه مان در CLI می رویم.

دستورات (Command)

با استفاده از دستور زیر ما می توانیم به محل پروژه مان برویم و لیست پروژه را ببینیم:

- PS C:\Users\Vignesh> cd '.\Documents\Visual Studio 2017\'
- PS C:\Users\Vignesh\Documents\Visual Studio 2017> cd '.\Projects'
- PS C:\Users\Vignesh\Documents\Visual Studio 2017\Projects> ls

با استفاده از "ls" می توانیم تمام لیست فایل ها یا پروژه ها را در مسیر مشخص شده مشاهده کنیم. ما می توانیم صفحه "Windows PowerShell" را شبیه تصویر زیر ببینیم.



```
PS C:\Users\Vignesh> cd '..\Documents\Visual Studio 2017\'
PS C:\Users\Vignesh\Documents\Visual Studio 2017> cd '..\Projects'
PS C:\Users\Vignesh\Documents\Visual Studio 2017\Projects> ls

Directory: C:\Users\Vignesh\Documents\Visual Studio 2017\Projects

Mode                LastWriteTime         Length Name
----                -
d-----          7/23/2017  10:33:37 PM             AngularJSaddUS
d-----          3/30/2017   9:46:34 PM             AngularJSBegin
d-----          7/3/2017    9:37:30 PM             DBFirstTest
d-----          8/11/2017  11:47:18 PM             DemoASP.NETCore
d-----          7/4/2017   10:01:35 PM             EFCore
d-----          3/9/2017    9:02:38 PM             First
d-----          3/22/2017   9:50:00 PM             FirstMVC6
d-----          7/19/2017   9:08:15 PM             Test
d-----          7/19/2017  11:04:40 PM             Test1
d-----          3/20/2017   9:39:30 PM             WebApplication1
d-----          3/20/2017   9:54:21 PM             WebApplication2
d-----          3/20/2017   9:56:35 PM             WebApplication3
d-----          3/26/2017   4:16:58 PM             WebApplication4
d-----          3/26/2017   8:23:38 PM             WebApplication5
d-----          3/27/2017   8:58:35 PM             WebApplication6
d-----          6/26/2017   9:17:15 PM             WebApplication7
d-----          7/23/2017   3:47:48 PM             WebApplication8
d-----          7/23/2017   4:03:52 PM             WebApplication9

PS C:\Users\Vignesh\Documents\Visual Studio 2017\Projects>
```

ما قصد داریم یک "class library" جدید در راه حل "DemoASP.NETCore" اضافه کنیم، اکنون به محل پروژه بروید و می توانید راه حل های "DemoASP.NETCore.sln" را ببینید.

دستورات (Command)

- PS C:\Users\Vignesh\Documents\Visual Studio 2017\Projects> cd
.\DemoASP.NETCore\

- PS C:\Users\Vignesh\Documents\Visual Studio 2017\Projects\DemoASP.NETCore> ls

```

Windows PowerShell
PS C:\Users\Vignesh\Documents\Visual Studio 2017\Projects> cd .\DemoASP.NETCore\
PS C:\Users\Vignesh\Documents\Visual Studio 2017\Projects\DemoASP.NETCore> ls

Directory: C:\Users\Vignesh\Documents\Visual Studio 2017\Projects\DemoASP.NETCore

Mode                LastWriteTime         Length Name
----                -
d-----          9/15/2017  8:44:42 PM             DemoASP.NETCore
-a-----          9/14/2017  10:47:34 PM          1012 DemoASP.NETCore.sln

PS C:\Users\Vignesh\Documents\Visual Studio 2017\Projects\DemoASP.NETCore> ls

```

فرمان HELP در CLI

با استفاده از دستور help ما میتوانیم اطلاعات کامل درباره یک فرمان خاص را در CLI بدست آوریم، به عنوان مثال، ما قصد داریم آیتم های جدید در NET Core را اضافه کنیم، بنابراین باید درباره "new" بدانیم. فرمان زیر را تایپ کنید تا در مورد "new" بدانید.

دستورات (Command)

- PS C:\Users\Vignesh\documents\Visual Studio 2017\Projects\DemoASP.NETCore> dotnet new -help

ما می توانیم با استفاده از دستور "dotnet new" با نام کوتاه (short name) نام قالب در CLI پروژه های (Console Application, Class Library, Unit Test Project, xUnit Test Project, ASP.NET) ایجاد کنیم. ما می توانیم مثال زیر را ببینیم:

Template	Short Name	Example Command
Console Application	console	dotnet new console
Class library	classlib	dotnet new classlib
Unit Test Project	mstest	dotnet new mstest
xUnit Test Project	xunit	dotnet new xunit
ASP.NET Core Empty	web	dotnet new web
ASP.NET Core Web App	mvc	dotnet new mvc
ASP.NET Core Web API	webapi	dotnet new webapi
Solution File	sln	dotnet new sln

```

Windows PowerShell
PS C:\Users\Uignesh\documents\Visual Studio 2017\Projects\DemoASP.NETCore> dotnet new --help
Getting ready...
Template Instantiation Commands for .NET Core CLI.

Usage: dotnet new [arguments] [options]

Arguments:
  template  The template to instantiate.

Options:
  -l|--list           List templates containing the specified name.
  -lang|--language   Specifies the language of the template to create
  -n|--name          The name for the output being created. If no name is specified, the name
                    used.
  -o|--output        Location to place the generated output.
  -h|--help          Displays help for this command.
  -all|--show-all   Shows all templates

Templates
-----
| Template                | Short Name | Language | Tags |
|-----|-----|-----|-----|
| Console Application    | console   | [C#], F# | Common/Console |
| Class library          | classlib  | [C#], F# | Common/Library  |
| Unit Test Project     | mstest    | [C#], F# | Test/MSTest     |
| xUnit Test Project    | xunit     | [C#], F# | Test/xUnit      |
| ASP.NET Core Empty    | web       | [C#]     | Web/Empty       |
| ASP.NET Core Web App  | mvc       | [C#], F# | Web/MVC         |
| ASP.NET Core Web API  | webapi    | [C#]     | Web/WebAPI      |
| Solution File         | sln       |          | Solution        |

Examples:
  dotnet new mvc --auth None --framework netcoreapp1.1
  dotnet new mvc --framework netcoreapp1.1
  dotnet new --help
PS C:\Users\Uignesh\documents\Visual Studio 2017\Projects\DemoASP.NETCore>

```

اضافه کردن Class Library جدید

ما قصد داریم با استفاده از فرمان CLI class library اضافه کنیم. ما می توانیم دستور زیر را برای اضافه کردن class library ببینیم. ما باید دستور زیر را در مسیر پروژه مشخص شده اجرا کنیم.

دستورات (Command)

- dotnet new classlib -n democlass -o democlass


```
Windows PowerShell
PS C:\Users\Uignesh\My Documents\Visual Studio 2017\Projects\DenoASP.NETCore> dotnet --help
.NET Command Line Tools (1.0.0)
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):
  -d|--diagnostics   Enable diagnostic output
  --version           Display .NET CLI Version Number
  --info             Display .NET CLI Info

Commands:
  new                Initialize .NET projects.
  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.
  migrate            Migrates a project.json based project to a msbuild based project.
  sln                Modify solution (SLN) files.

Project modification commands:
  add                Add items to the project
  remove             Remove items from the project
  list               List items in the project

Advanced Commands:
  nuget              Provides additional NuGet commands.
  msbuild            Runs Microsoft Build Engine (MSBuild).
  vstest             Runs Microsoft Test Execution Command Line Tool.
PS C:\Users\Uignesh\My Documents\Visual Studio 2017\Projects\DenoASP.NETCore>
```

پس از اضافه کردن class library پیامی را به عنوان (The template "Class library" created successfully) دریافت می کنیم. ما می توانیم تمام لیست آیتم ها را در solutions مان با استفاده از دستور "ls" را ببینید.

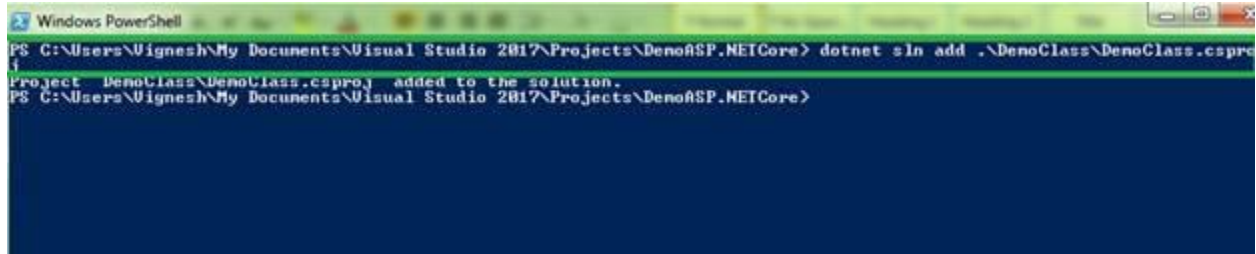
تغییر Solution (Modify the Our Solution)

ما قصد داریم Solution ها را تغییر بدهیم تا فرمان را برای تغییر Solution ها بدانیم. ما از دستور "dotnet --help" استفاده می کنیم. هنگام ورود به دستور "dotnet -help" می توانیم تصویر زیر را ببینیم و می توانیم دستور را برای تغییر Solution ها ببینیم.

ما از دستور "sln" برای تغییر solutions ها استفاده می کنیم. ما می توانیم تصویر بالا را ببینیم. ما باید دستور زیر را اجرا کنیم در پروژه مان که مسیر را مشخص می کند.

دستورات (Command)

- `dotnet sln add .\DemoClass\DemoClass.csproj`



```

Windows PowerShell
PS C:\Users\Wignesh\My Documents\Visual Studio 2017\Projects\DemoASP.NETCore> dotnet sln add .\DemoClass\DemoClass.csproj
Project DemoClass\DemoClass.csproj added to the solution.
PS C:\Users\Wignesh\My Documents\Visual Studio 2017\Projects\DemoASP.NETCore>

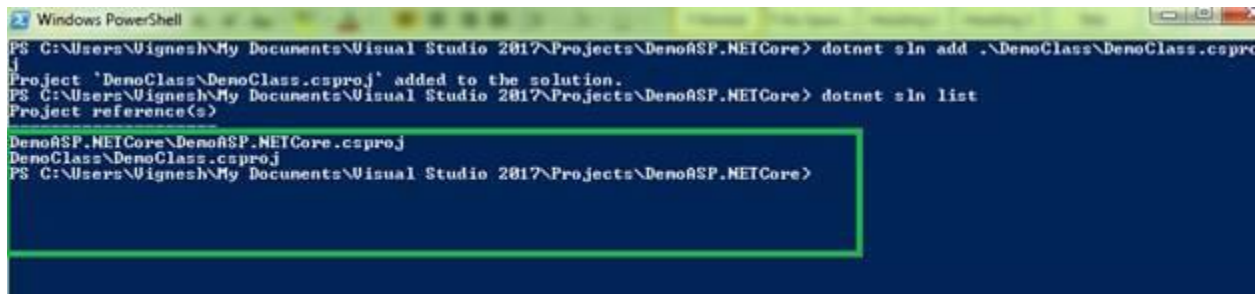
```

پس از اجرای دستور ما پیامی را به عنوان (") "Project `DemoClass\DemoClass.csproj` added to the solution." دریافت می کنیم.

حالا ما می توانیم همه پروژه ها را در راه حل هایمان با استفاده از دستور زیر CLI ببینیم.

دستورات (Command)

- `dotnet sln list`



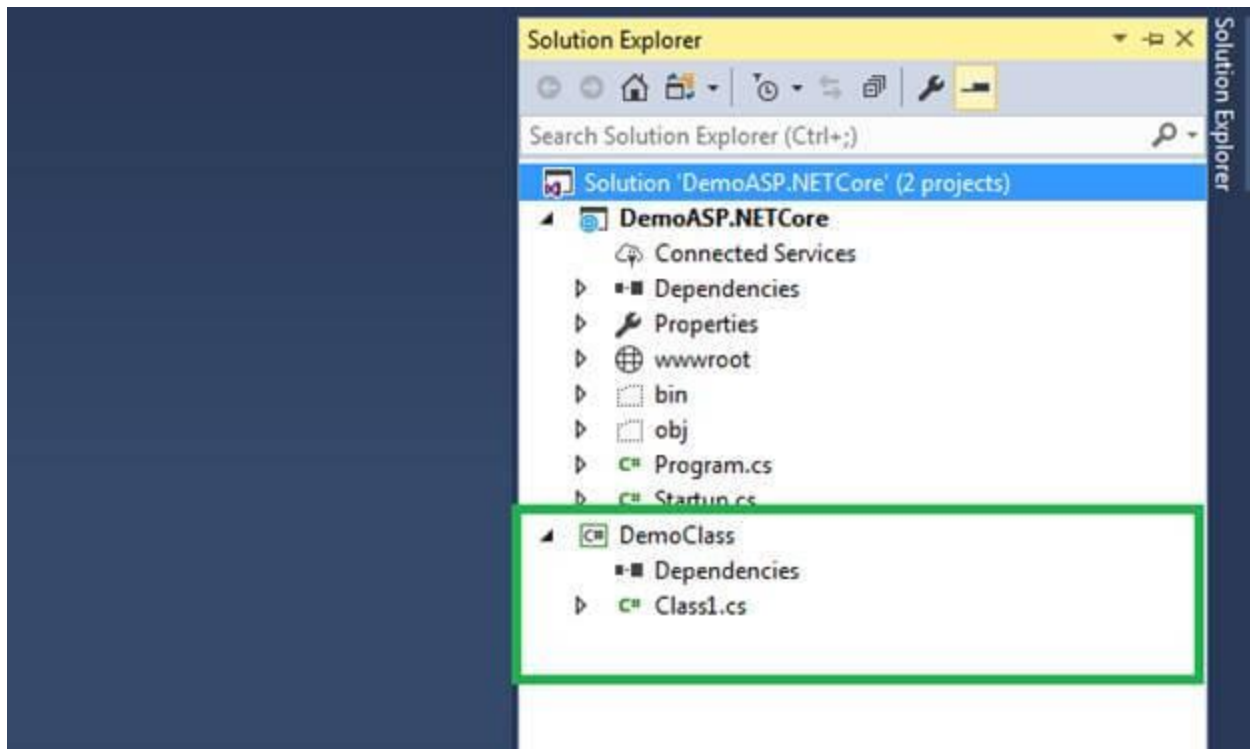
```

Windows PowerShell
PS C:\Users\Wignesh\My Documents\Visual Studio 2017\Projects\DemoASP.NETCore> dotnet sln add .\DemoClass\DemoClass.csproj
Project 'DemoClass\DemoClass.csproj' added to the solution.
PS C:\Users\Wignesh\My Documents\Visual Studio 2017\Projects\DemoASP.NETCore> dotnet sln list
Project reference(s)
DemoASP.NETCore\DemoASP.NETCore.csproj
DemoClass\DemoClass.csproj
PS C:\Users\Wignesh\My Documents\Visual Studio 2017\Projects\DemoASP.NETCore>

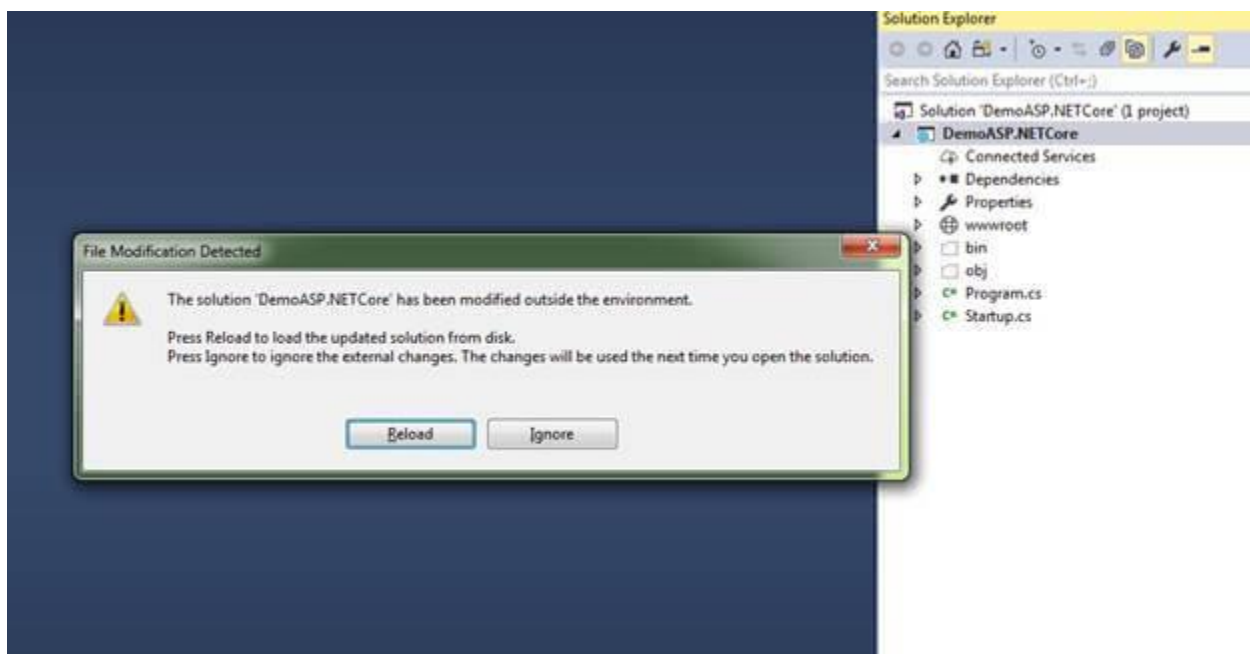
```

دو پروژه در solutions های ما وجود دارد. یکی از آنها "Demo ASP.NET Core.csproj" است و یکی دیگر از آنها "Demo Class.csproj" است.

حال ما به ویژوال استودیو می رویم و پیامی به عنوان "The Solutions DemoASP.NETCore has been modified outside the environment (سولوشن DemoASP.NETCore در خارج از این محیط تغییر یافت است) می بینیم. ما می توانیم پیام کامل را در تصویر بالا ببینیم. هنگامی که ما دکمه "reload" را فشار می دهیم، solution ما دوباره بارگذاری می شود و می توانیم "class library" جدیدی که ما اضافه کردیم را ببینیم. اکنون می توانیم class library که اضافه کردیم در تصویر زیر ببینید:



ما می توانیم "reference" را به هر پروژه ای اضافه کنیم و از "class library" استفاده کنیم.



نتیجه

این مقاله نحوه اضافه کردن "class library" را با استفاده از دستور "CLI" به صورت گام به گام توضیح می دهد.