
.NET Conf China
2022
线上+线下 2022.12.03-04
开源 · 安全 · 赋能

.NET Conf China

2022

dotnet-exec 让 C# 更简单

李卫涵
iHerb .NET 开发工程师/微软 MVP

Microsoft® Most Valuable Professional

What’s it

- dotnet tool (dotnet tool update –g dotnet-execute)
- Implicit project
- 自定义入口方法
- 远程代码执行
- 原始代码执行

How it works

获取代码

代码编译

代码执行

How it works

Console

Main

DLL

Custom Entry

Execute

Script

How to use

- dotnet tool

```
dotnet tool update -g dotnet-execute
```

```
dotnet tool update -g dotnet-execute --prerelease
```

How to use

- Docker

```
docker run --rm weihanli/dotnet-exec:latest dotnet-exec "Guid.NewGuid()"
```

```
docker run --rm --pull=always weihanli/dotnet-exec:latest
dotnet-exec "ApplicationHelper.RuntimeInfo"
```

How to use


```
PS C:\projects\sources\dotnet-exec> dotnet-exec --help
Description:

Usage:
  dotnet-exec <script> [command] [options]

Arguments:
  <script>  CSharp script to execute

Options:
  -f, --framework <net6.0|net7.0>      Target framework [default: net7.0]
  --startup-type <startup-type>        The startup type to use for finding the correct entry
  --entry <entry>                      Custom entry point('MainTest' by default)
  --compiler-type <default|workspace>  The compiler to use [default: default]
  --executor-type <default>            The executor to use [default: default]
  --preview                            Use preview language feature and enable preview features
  -c, --configuration <Debug|Release>  Compile configuration/OptimizationLevel
  --args, --arguments <arguments>      Input arguments
  --debug                              Enable debug logs for debug
  --ref-compile                        Use Ref assemblies for compile, when not found from local download from nuget
  --project <project>                  The project file path to exact references and usings
  --wide                               Includes widely-used
                                       references(Microsoft.Extensions.Configuration/DependencyInjection/Logging,Newtonsoft.Json,WeihanLi.Co
                                       mmon) [default: True]
  -w, --web                            Includes Web SDK references
  -r, --reference <reference>          Additional references
  -u, --using <using>                  Namespace usings
  --ad, --addition <addition>          Additional script path
  --profile <profile>                  The config profile to use
  --version                            Show version information
  -?, -h, --help                       Show help and usage information


Commands:
  profile  Configure user config profile
```

Code source

- Local file 本地文件

```
PS C:\projects\sources\SamplesInPractice\CSharp11Sample> cat .\NameOfSample.cs
using System.ComponentModel.DataAnnotations;
using System.Reflection;
using System.Runtime.CompilerServices;

namespace CSharp11Sample;
internal static class NameOfSample
{
    [Display(Name = nameof(MainTest))]
    public static void MainTest()
    {
        var displayName = MethodBase.GetCurrentMethod()
            ?.GetCustomAttribute<DisplayAttribute>()
            ?.Name;
        Console.WriteLine(displayName);

        Hello(1 + 1 > 2);
    }

    private static void Hello(bool condition, [CallerArgumentExpression(nameof(condition))] string? expression = null)
    {
        Console.WriteLine($"{expression} : {condition}");
    }
}
PS C:\projects\sources\SamplesInPractice\CSharp11Sample> dotnet-exec .\NameOfSample.cs
MainTest
1 + 1 > 2 : False
```

Code source

- Remote file 远程文件

```
PS C:\projects\sources\SamplesInPractice\CSharp11Sample> dotnet-exec https://github.com/WeihanLi/SamplesInPractice/blob/master/CSharp11Sample/NameOfSample.cs
MainTest
1 + 1 > 2 : False
PS C:\projects\sources\SamplesInPractice\CSharp11Sample> dotnet-exec https://raw.githubusercontent.com/WeihanLi/SamplesInPractice/master/CSharp11Sample/NameOfSample.cs
MainTest
1 + 1 > 2 : False
```

```
PS C:\projects\sources\dotnet-exec> dotnet-exec https://gist.github.com/WeihanLi/7b4032a32f1a25c5f2d84b6955fa83f3
Current date:  2022-11-29
```

dotnet-exec test script

print-date.cs — Raw

```
1  Console.WriteLine($"Current date: {DateTime.Now.Date: yyyy-MM-dd}");
```

WeihanLi 提交于 刚刚 . add test/test.cs.

```
1  Console.WriteLine("Hello world");
```

PowerShell

```
PS C:\projects\sources\dotnet-exec> dotnet-exec https://gitee.com/weihanli/storage/blob/master/test/test.cs
Hello world
```

Raw code execution

```
PS C:\projects\sources\SamplesInPractice> dotnet-exec 'Console.WriteLine("Hello world!");'
Hello world!
```

```
PS C:\projects\sources\SamplesInPractice> dotnet-exec 'Guid.NewGuid()'
[d3b83962-bbae-4f6a-a62b-926376295080]
```

```
PS C:\projects\sources\SamplesInPractice> dotnet-exec 'WebApplication.Create().Run();' -w
info: Microsoft.Hosting.Lifetime[14]
      Now listening on: http://localhost:5000
info: Microsoft.Hosting.Lifetime[0]
      Application started. Press Ctrl+C to shut down.
info: Microsoft.Hosting.Lifetime[0]
      Hosting environment: Production
info: Microsoft.Hosting.Lifetime[0]
      Content root path: C:\projects\sources\SamplesInPractice
info: Microsoft.Hosting.Lifetime[0]
      Application is shutting down...
```

Usings

- Namespace using

```
PS C:\projects\sources\SamplesInPractice> dotnet-exec 'JsonConvert.SerializeObject(new{Name="Xiaoming"})' --using Newton
soft.Json
"{\"Name\":\"Xiaoming\"}"
```

- Static using

```
PS C:\projects\sources\SamplesInPractice> dotnet-exec 'SerializeObject(new{Name="Xiaoming"})' --using 'static Newtonsoft
.Json.JsonConvert'
"{\"Name\":\"Xiaoming\"}"
```

- Alias using

```
PS C:\projects\sources\SamplesInPractice> dotnet-exec 'MyJson.SerializeObject(new{Name="Xiaoming"})' --using 'MyJson = N
ewtonsoft.Json.JsonConvert'
"{\"Name\":\"Xiaoming\"}"
```

References

- Local file dll reference

```
PS C:\projects\sources\dotnet-exec\tests\IntegrationTest\bin\Debug\net7.0> dotnet-exec 'ApplicationHelper.GetLibraryInfo
(typeof(WeihanLi.Npoi.CsvHelper))' --reference ".\out\WeihanLi.Npoi.dll"
LibraryInfo { LibraryHash="4875c1619ff4ae811411cb0ef2c4d93273426fa5", LibraryVersion=[2.0.0], RepositoryUrl="https://git
hub.com/WeihanLi/WeihanLi.Npoi" }
```

- Local folder dll reference

```
PS C:\projects\sources\dotnet-exec\tests\IntegrationTest\bin\Debug\net7.0> dotnet-exec 'ApplicationHelper.GetLibraryInfo
(typeof(WeihanLi.Npoi.CsvHelper))' --reference "folder: .\out"
LibraryInfo { LibraryHash="4875c1619ff4ae811411cb0ef2c4d93273426fa5", LibraryVersion=[2.0.0], RepositoryUrl="https://git
hub.com/WeihanLi/WeihanLi.Npoi" }
```

References

- NuGet package reference

```
PS C:\projects\sources\SamplesInPractice> dotnet-exec 'ApplicationHelper.GetLibraryInfo(typeof(WeihanLi.Npoi.CsvHelper))
' --reference "nuget: WeihanLi.Npoi"
LibraryInfo { LibraryHash="8e2c1dee6efee9b7b4b12f16272f266c9ad09233", LibraryVersion=[2.4.2], RepositoryUrl="https://git
hub.com/WeihanLi/WeihanLi.Npoi" }
PS C:\projects\sources\SamplesInPractice> dotnet-exec 'ApplicationHelper.GetLibraryInfo(typeof(WeihanLi.Npoi.CsvHelper))
' --reference "nuget: WeihanLi.Npoi, 2.4.2"
LibraryInfo { LibraryHash="8e2c1dee6efee9b7b4b12f16272f266c9ad09233", LibraryVersion=[2.4.2], RepositoryUrl="https://git
hub.com/WeihanLi/WeihanLi.Npoi" }
```

- Project reference

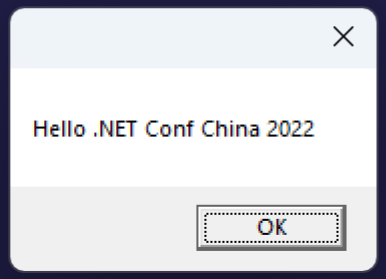
```
PS C:\projects\sources\dotnet-exec> dotnet-exec 'ApplicationHelper.GetLibraryInfo(typeof(Csv
Helper))' --reference 'project:C:\projects\sources\WeihanLi.Npoi\src\WeihanLi.Npoi\WeihanLi.
Npoi.csproj' --using 'WeihanLi.Npoi'
LibraryInfo { LibraryHash="c1215adc1fe47554d9ef7f996117bb45ad067dc4", LibraryVersion=[2.4.2]
, RepositoryUrl="https://github.com/WeihanLi/WeihanLi.Npoi" }
```

References

- Framework reference

```
PS C:\projects\sources\dotnet-exec> dotnet-exec 'WebApplication.Create().Run();' --reference 'framework:web'
info: Microsoft.Hosting.Lifetime[14]
      Now listening on: http://localhost:5000
info: Microsoft.Hosting.Lifetime[0]
      Application started. Press Ctrl+C to shut down.
info: Microsoft.Hosting.Lifetime[0]
      Hosting environment: Production
info: Microsoft.Hosting.Lifetime[0]
      Content root path: C:\projects\sources\dotnet-exec
```

```
PS C:\projects\sources\dotnet-exec> dotnet-exec 'MessageBox.Show("Hello .NET Conf China 2022");' --reference
'framework:WindowsDesktop'
```



Reference && using from project file


```
PS C:\projects\sources\dotnet-exec> dotnet-exec 'WriteLine(MyFile.Exists("appsettings.json"));' --project
'https://raw.githubusercontent.com/WeihanLi/SamplesInPractice/master/net6sample/ImplicitUsingsSample/Impli
citUsingsSample.csproj'
False
```

```
<Project Sdk="Microsoft.NET.Sdk">

  <PropertyGroup>
    <OutputType>Exe</OutputType>
    <TargetFramework>net6.0</TargetFramework>
    <ImplicitUsings>enable</ImplicitUsings>
    <Nullable>enable</Nullable>
  </PropertyGroup>

  <ItemGroup>
    <PackageReference Include="WeihanLi.Common" Version="1.0.46" />
  </ItemGroup>

  <ItemGroup>
    <Using Include="System.Console" Static="true" />
    <Using Include="WeihanLi.Common.Helpers" />
    <Using Include="System.IO.File" Alias="MyFile" />
    <Using Remove="System" />
  </ItemGroup>
</Project>
```

Config profile

```
PS C:\projects\sources\dotnet-exec> dotnet-exec profile --help
Description:
  Configure user config profile

Usage:
  dotnet-exec <script> profile [command] [options]

Arguments:
  <script>  CSharp script to execute

Options:
  -?, -h, --help  Show help and usage information


Commands:
  set <profile-name>  Configure config profile
  get <profile-name>  Get config profile
  rm <profile-name>   Remove config profile
  ls                  List the config profiles configured
```

Config profile

```
PS C:\projects\sources\dotnet-exec> dotnet-exec profile set web -r "nuget:WeihanLi.Web.Extensions" -u 'WeihanLi.Web.Extensions' --web --wide true
PS C:\projects\sources\dotnet-exec> dotnet-exec profile get web
{
  "Usings": [
    "WeihanLi.Web.Extensions"
  ],
  "References": [
    "nuget:WeihanLi.Web.Extensions"
  ],
  "IncludeWebReferences": true,
  "IncludeWideReferences": true,
  "EntryPoint": null,
  "EnablePreviewFeatures": false
}
PS C:\projects\sources\dotnet-exec> dotnet-exec profile ls
Profiles:
- web
```

```
PS C:\projects\sources\dotnet-exec> dotnet-exec profile rm web
PS C:\projects\sources\dotnet-exec> dotnet-exec profile ls
No profiles found
```

Config profile

```
PS C:\projects\sources\dotnet-exec> dotnet-exec 'WebApplication.Create().Chain(_⇒_.MapRunti
meInfo()).Run();' --profile web
info: Microsoft.Hosting.Lifetime[14]
      Now listening on: http://localhost:5000
info: Microsoft.Hosting.Lifetime[0]
      Application started. Press Ctrl+C to shut down.
info: Microsoft.Hosting.Lifetime[0]
      Hosting environment: Production
info: Microsoft.Hosting.Lifetime[0]
      Content root path: C:\projects\sources\dotnet-exec
info: Microsoft.AspNetCore.Hosting.Diagnostics[1]
      Request starting HTTP/1.1 GET http://localhost:5000/runtime-info - -
info: Microsoft.AspNetCore.Routing.EndpointMiddleware[0]
      Executing endpoint 'HTTP: GET /runtime-info'
info: Microsoft.AspNetCore.Routing.EndpointMiddleware[1]
      Executed endpoint 'HTTP: GET /runtime-info'
info: Microsoft.AspNetCore.Hosting.Diagnostics[2]
      Request finished HTTP/1.1 GET http://localhost:5000/runtime-info - - - 200 - applicati
on/json;+charset=utf-8 211.4940ms
info: Microsoft.AspNetCore.Hosting.Diagnostics[1]
      Request starting HTTP/1.1 GET http://localhost:5000/runtime-info - -
info: Microsoft.AspNetCore.Routing.EndpointMiddleware[0]
      Executing endpoint 'HTTP: GET /runtime-info'
info: Microsoft.AspNetCore.Routing.EndpointMiddleware[1]
      Executed endpoint 'HTTP: GET /runtime-info'
info: Microsoft.AspNetCore.Hosting.Diagnostics[2]
      Request finished HTTP/1.1 GET http://localhost:5000/runtime-info - - - 200 - applicati
on/json;+charset=utf-8 2.8614ms
```

References

- https://github.com/WeihanLi/dotnet-exec
- https://www.nuget.org/packages/dotnet-execute
- https://hub.docker.com/r/weihanli/dotnet-exec
- https://github.com/dotnet/roslyn
- https://github.com/dotnet/command-line-api
- https://github.com/NuGet/NuGet.Client

.NET Conf China 2022
开源 · 安全 · 赋能

Thank you!