

# 教程：创建项目模板

2019/06/25 · 👤 👤

## 本文内容

[先决条件](#)

[创建项目模板](#)

[修改 Program.cs](#)

[修改 consoleasync.csproj](#)

[生成项目](#)

[创建模板配置](#)

[后续步骤](#)

使用 .NET Core，可以创建和部署可生成项目、文件甚至资源的模板。本教程是系列教程的第二部分，介绍如何创建、安装和卸载用于 `dotnet new` 命令的模板。

在本系列的这一部分中，你将了解如何：

- ✓ 创建项目模板的资源
- ✓ 创建模板配置文件夹和文件
- ✓ 从文件路径安装模板
- ✓ 测试项模板
- ✓ 卸载项模板

## 先决条件

- 完成本系列教程的[第 1 部分](#)。
- 打开终端并导航到 `working\templates` 文件夹。

## 创建项目模板

项目模板生成可立即运行的项目，使用户可以轻松地使用一组有效的代码。.NET Core 包含一些项目模板，例如控制台应用程序或类库。在本例中，你将创建一个新的控制台项目，该项目启用 C# 8.0 并生成 `async main` 入口点。

在终端中，导航到 `working\templates` 文件夹，并创建一个名为“`consoleasync`”的新子文件夹。进入子文件夹，并运行 `dotnet new console` 以生成标准控制台应用程序。将编辑此模板生成的文件以创建新模板。

```
console
```

 复制

```
working
├── templates
│   └── consoleasync
│       ├── consoleasync.csproj
│       └── Program.cs
```

## 修改 Program.cs

打开 program.cs 文件。控制台项目不使用异步入口点，我们来添加它。将代码更改为以下内容并保存文件：

C#

复制

```
using System;
using System.Threading.Tasks;

namespace consoleasync
{
    class Program
    {
        static async Task Main(string[] args)
        {
            await Console.Out.WriteLineAsync("Hello World with C# 8.0!");
        }
    }
}
```

## 修改 consoleasync.csproj

将项目使用的 C# 语言版本更新到 8.0 版。编辑 consoleasync.csproj 文件并将 <LangVersion> 设置添加到 <PropertyGroup> 节点。

XML

复制

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


  <PropertyGroup>
    <OutputType>Exe</OutputType>
    <TargetFramework>netcoreapp2.2</TargetFramework>

    <LangVersion>8.0</LangVersion>
  </PropertyGroup>

</Project>
```

# 生成项目

在完成项目模板之前，应对其进行测试，确保它能够正确编译和运行。在终端中，运行 `dotnet run` 命令，应能看到以下输出：

|  |  |
|--|--|
| console  |  复制 |
| <pre>C:\working\templates\consoleasync&gt; dotnet run Hello World with C# 8.0!</pre> |  |

可以使用 `dotnet run` 删除已创建的 `obj` 和 `bin` 文件夹。删除这些文件可确保你的模板仅包含与模板相关的文件，而不包含生成操作产生的任何文件。


现在你已经创建了模板的内容，需要在模板的根文件夹中创建模板配置。

## 创建模板配置

模板在 .NET Core 中通过模板根目录中的特殊文件夹和配置文件进行识别。在本教程中，你的模板文件夹位于 `working\templates\consoleasync`。

创建模板时，除特殊配置文件夹外，模板文件夹中的所有文件和文件夹都作为模板的一部分包含在内。此配置文件夹名为 `template.config`。

首先，创建一个名为 `template.config` 的新子文件夹，然后进入该文件夹。然后，创建一个名为 `template.json` 的新文件。文件夹结构应如下所示：

|  |  |
|--|--|
| console  |  复制 |
| <pre>working ├── templates │   └── consoleasync │       ├── .template.config │       └── template.json</pre> |  |

使用你喜爱的文本编辑器打开 `template.json` 并粘贴以下 json 代码，然后保存：

|  |  |
|--|--|
| JSON   |  复制 |
| <pre>{   "\$schema": "http://json.schemastore.org/template",   "author": "Me",   "classifications": [ "Common", "Console", "C#8" ],   "identity": "ExampleTemplate.AsyncProject",   "name": "Example templates: async project",   "shortName": "consoleasync",   "tags": {</pre> |  |

```

    "language": "C#",
    "type": "project"
  }
}
```

此配置文件包含模板的所有设置。可以看到基本设置，例如 `name` 和 `shortName`，除此之外，还有一个设置为 `project` 的 `tags/type` 值。这会将模板指定为项目模板。你创建的模板类型不存在限制。`item` 和 `project` 值是 .NET Core 建议使用的通用名称，便于用户轻松筛选正在搜索的模板类型。

classifications 项表示你在运行 `dotnet new` 并获取模板列表时看到的“标记”列。用户还可以根据分类标记进行搜索。不要将 json 文件中的 tags 属性与 classifications 标记列表混淆。它们虽然具有类似的名称，但截然不同。template.json 文件的完整架构位于 [JSON 架构存储](#)。有关 template.json 文件的详细信息，请参阅 [dotnet 创建模板 wiki](#)。

现在你已有一个有效的 `.template.config/template.json` 文件，可以安装模板了。在安装模板之前，请务必删除无需在模板中包含的任何额外文件夹和文件，例如 `bin` 或 `obj` 文件夹。在终端中，导航到 `consoleasync` 文件夹，并运行 `dotnet new -i .\` 以安装位于当前文件夹的模板。如果使用的是 Linux 或 MacOS 操作系统，请使用正斜杠：`dotnet new -i ./`。

此命令输出安装的模板列表，其中应包括你的模板。

复制

```
C:\working\templates\consoleasync> dotnet new -i .\
Usage: new [options]
```

Options:

```
-h, --help      Displays help for this command.
-l, --list      Lists templates containing the specified name. If no
name is specified, lists all templates.
```

```
... cut to save space ...
```

| Templates | Short Name |
|-----------|------------|
|-----------|------------|

Language      Tags

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

|                     |         |
|---------------------|---------|
| Console Application | console |
|---------------------|---------|

|              |                |
|--------------|----------------|
| [C#], F#, VB | Common/Console |
|--------------|----------------|

Example templates: `async project` `consoleasync` `[C#]`

Common/Console/C#8

|               |          |
|---------------|----------|
| Class library | classlib |
|---------------|----------|

| [C#], F#, VB | Common/Library |
|--------------|----------------|
|              |                |


```
WPF Application                                     wpf
```


|          |            |
|----------|------------|
| [C#], VB | Common/WPF |
|----------|------------|

|                                      |                 |      |
|--------------------------------------|-----------------|------|
| Windows Forms (WinForms) Application | winforms        |      |
| [C#], VB                             | Common/WinForms |      |
| Worker Service                       | worker          | [C#] |
| Common/Worker/Web                    |                 |      |

## 测试项目模板

现在你已安装了项模板，可对其进行测试。导航到 test 文件夹，使用 `dotnet new consoleasync` 创建新的控制台应用程序。这将生成一个可以使用 `dotnet run` 命令轻松测试的工作项目。


|  |  |
|--|--|
| console  |  复制 |
| <pre>C:\test&gt; dotnet new consoleasync The template "Example templates: async project" was created successfully.</pre> |  |

|  |  |
|--|--|
| console  |  复制 |
| <pre>C:\test&gt; dotnet run Hello World with C# 8.0!</pre> |  |

祝贺你！你已使用 .NET Core 创建并部署了项目模板。为准备学习本系列教程的下一部分，必须卸载已创建的模板。确保同时删除 test 文件夹中的所有文件。这将回到干净状态，为本教程的下一个主要部分做好准备。

## 卸载模板

由于模板是使用文件路径安装的，因此，必须使用绝对文件路径将其卸载。可以通过运行 `dotnet new -u` 命令看到已安装的模板列表。你的模板应列在最后。使用列出的路径，通过执行 `dotnet new -u <ABSOLUTE PATH TO TEMPLATE DIRECTORY>` 命令卸载模板。

|  |  |
|--|--|
| console  |  复制 |
| <pre>C:\working&gt; dotnet new -u Template Instantiation Commands for .NET Core CLI  Currently installed items:   Microsoft.DotNet.Common.ItemTemplates     Templates:       dotnet gitignore file (gitignore)       global.json file (globaljson)       NuGet Config (nugetconfig)       Solution File (sln)       Dotnet local tool manifest file (tool-manifest)       Web Config (webconfig)</pre> |  |

... cut to save space ...

NUnit3.DotNetNew.Template

Templates:


NUnit 3 Test Project (nunit) C#  
NUnit 3 Test Item (nunit-test) C#  
NUnit 3 Test Project (nunit) F#  
NUnit 3 Test Item (nunit-test) F#  
NUnit 3 Test Project (nunit) VB  
NUnit 3 Test Item (nunit-test) VB

C:\working\templates\consoleasync

Templates:

Example templates: async project (consoleasync) C#

console

 复制



C:\working> dotnet new -u C:\working\templates\consoleasync

## 后续步骤

在本教程中，你创建了一个项目模板。 若要了解如何将项模板和项目模板打包为易于使用的文件，请继续学习本教程系列。

创建模板包

此页面有帮助吗？

 是  否