**.NET CLI Cheat Sheet**

[ES](https://abarrenechea.net/articles/es/comandos-mas-usados-de-dot-net-cli)

The .NET Command Line Interface (CLI) is a cross-platform tool that can create, build, run, and publish .NET projects. For more information about it, visit the official documentation [site](https://docs.microsoft.com/en-us/dotnet/core/tools/)

It can be handy in cases such as:

* Environments where you don’t have an IDE, such as Visual Studio
* Automation tasks using scripts
* Environments where a UI is not available

**Get help**

**Get help related to all available dotnet commands**

dotnet --help

**Get help related to a specific dotnet command**

Get help about a specific command, replace [command-name] by an actual command.

dotnet [command-name] --help

Example: Get help about dotnet new command.

dotnet new --help

**Check versions available**

**Check dotnet version**

Will display the dotnet SDK version the CLI will use. The last version available will be used in case multiple versions are installed.

dotnet --version

**List SDKs**

List available Software Develpment Kits (SDK) available in the machine. The CLI will use the latest version found by default.

dotnet --list-sdks

**List runtimes**

List available .NET Core runtimes.

dotnet --list-runtimes

**Target a specific SDK**

In some cases you may want to target a specific SDK version different from the latest one. For this purpose you can specify the version in a globaljson file.

dotnet new globaljson --sdk-version [version]

**List available templates**

List available templates; the list will vary depending on the current SDK.

dotnet new --list

**Development Cycle**

**Create a new Solution**

Creates a new empty solution in the folder where the command executed.

dotnet new sln -n [solution-name] -o [output-folder]

**Adds gitignore file**

Adds gitignore, this command should be run in the root folder of your git repository.

dotnet new gitignore

**Create a new Project**

Creates a new project in the path specified. This command will only create the project, you still need to add it to a solution.

dotnet new [template-name] -n [project-name] -o [output-folder]

**Add a Project to a Solution**

Adds the project to the specified solution. Run this command from the solution folder.

dotnet sln add [path to the csproj to add]

**Add a nuget package to a project**

Add a nuget package.

dotnet add [project] package [package-name] --version [version-number]

**Build**

Builds as solution.

dotnet build

**Run**

Run an application, automatically runs dotnet build.

dotnet run --project [path-to-the-project]

**Publish**

Publish an application.

dotnet publish -r win-x64 -c Release -o [output-path]

**Examples**

**Creating a Console App with .NET CLI**

* Set the SDK version to 5.0.403
* Creates a new foler for the solution
* Creates a new solution
* Navigates to the Solution folder
* Creates a folder for the source files
* Creates a new Project using the "console" template
* Add the new Project to the Solution

dotnet new globaljson --sdk-version 5.0.403  
md ConsoleApp  
dotnet new sln -n ConsoleApp -o ConsoleApp  
cd ConsoleApp  
md src  
dotnet new console -n ConsoleApp -o src/ConsoleApp  
dotnet sln add src/consoleapp/consoleapp.csproj  
dotnet build  
dotnet run --project src/consoleapp

**Creating a MVC App with .NET CLI**

1. Creates a new foler for the solution
2. Creates a new solution in the folder created in the previous step
3. Navigates to the Solution folder
4. Creates a folder for the source files
5. Creates a new mvc app
6. Adds the new Project to the Solution
7. Builds the solution
8. Runs the app

md MvcApp  
dotnet new sln -n MvcApp -o MvcApp  
cd MvcApp  
md src  
dotnet new mvc -uld --auth individual -n MvcApp -o src/MvcApp  
dotnet sln add src/MvcApp/MvcApp.csproj  
dotnet build  
dotnet run --project src/MvcApp/MvcApp.csproj

**Creating a Web API with .NET CLI**

1. Creates a new foler for the solution
2. Creates a new solution in the folder created in the previous step
3. Navigates to the Solution folder
4. Creates a folder for the source files
5. Creates a new Web API
6. Adds the new Project to the Solution
7. Builds the solution
8. Runs the Web API

md WebApi  
dotnet new sln -n WebApi -o WebApi  
cd WebApi  
md src  
dotnet new webapi -n WebApi -o src/WebApi --use-program-main true  
dotnet sln add src/WebApi/WebApi.csproj  
dotnet build  
dotnet run --project src/WebApi/WebApi.csproj