.NET CLI Cheat Sheet

The .NET command-line interface (CLI) is a cross-platform toolchain for developing, building, running, and publishing .NET applications.

Displaying environment information	
dotnet --info	Prints out detailed information about your .NET installation
dotnet --version	Prints out the version of the .NET SDK used by dotnet commands

Creating new projects	
dotnet new –list	Lists available templates
dotnet new webapi -n Play.Catalog -f net5.0	Creates an ASP.NET Core 5 REST API in a new Play.Catalog directory
dotnet new classlib -n Play.Common -f net5.0	Creates a .NET 5 Class Library in a new Play.Common directory

Building projects	
dotnet build	Builds the project in the current directory and its dependencies
dotnet build -c Release	Build a project using the Release configuration
dotnet build --no-restore	Builds a project without restoring its dependencies

Running projects	
dotnet run	Runs the project in the current directory
dotnet run --project src\Play.Catalog.Service	Runs the project in the src\Play.Catalog.Service directory
dotnet run --no-build	Runs a project without first building it

Adding project/package references	
dotnet add reference ..\Play.Catalog.Contracts\Play.Catalog.Contracts.csproj	Adds a Play.Catalog.Contracts.csproj project reference to the project in the current directory
dotnet add package MassTransit	Adds a MassTransit NuGet package reference to the project in the current directory
dotnet add package MassTransit --version 8.0.3	Specifies the version to use for the NuGet package reference

Creating NuGet packages for local usage	
dotnet pack	Packs the project in the current directory into a NuGet package
dotnet pack -o D:\projects\packages	Places the generated package in the D:\projects\packages directory
dotnet pack -p:PackageVersion=1.0.1	Assigns version 1.0.1 to the generated package
dotnet pack -c Release	Packs the project using the Release configuration
dotnet nuget add source D:\projects\packages -n PlayEconomy	Adds the D:\projects\packages directory as a new NuGet source named PlayEconomy

Creating NuGet packages for GitHub usage	
dotnet pack -p:RepositoryUrl=https://github.com/dotnetmicroservices/play.common	Sets the RepositoryUrl property for the generated package
dotnet nuget add source --username USERNAME --password <PAT> --store-password-in-clear-text --name github "https://nuget.pkg.github.com/dotnetmicroservices/index.json"	Adds a GitHub Packages NuGet source named github using the specified personal access token
dotnet nuget push Play.Common.1.0.1.nupkg --api-key <PAT> --source "github"	Publishes the specified package to GitHub Packages