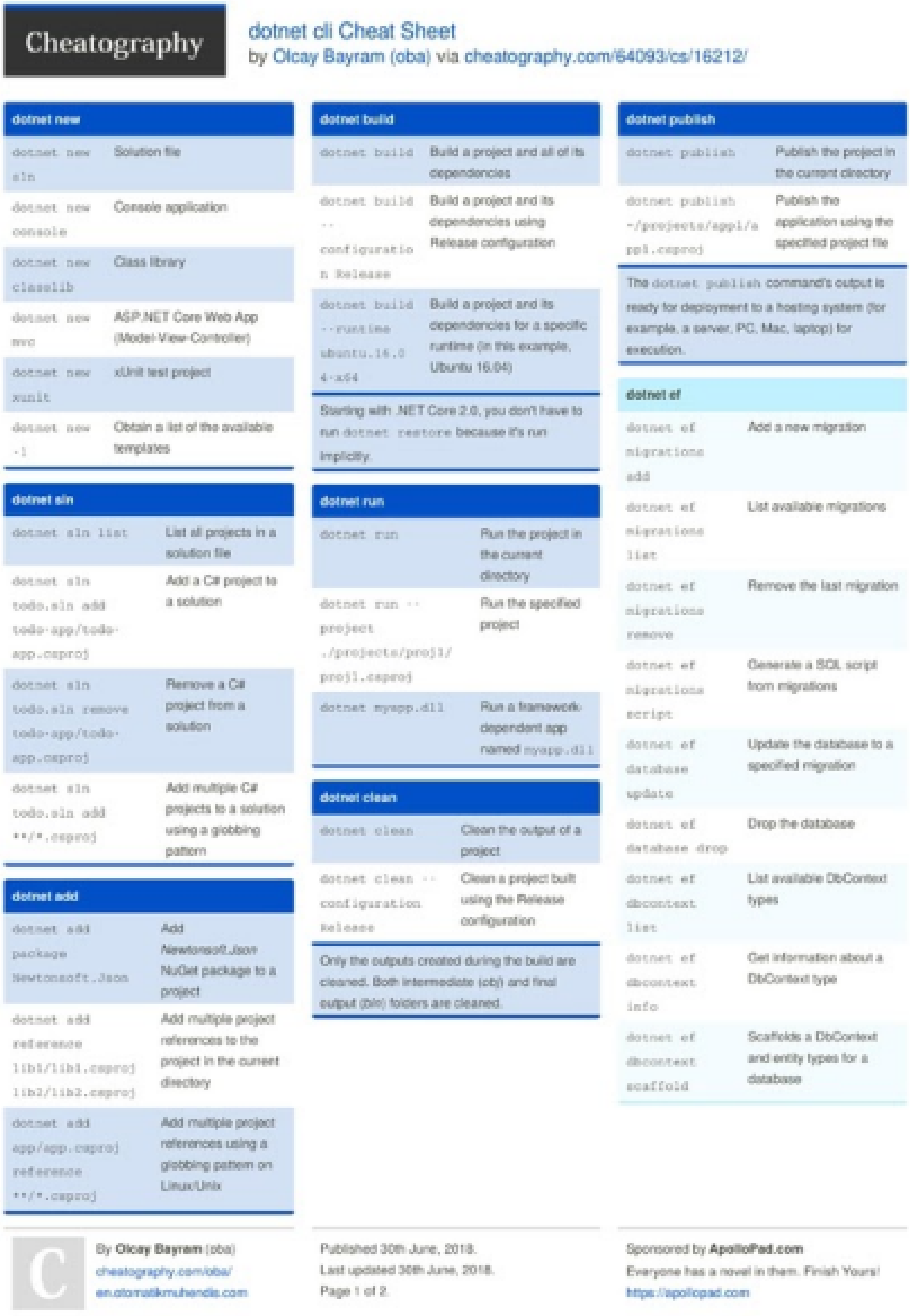

Cheatography

dotnet cli Cheat Sheet

by Olcay Bayram (oba) via cheatography.com/64093/cs/16212/

dotnet new

`dotnet new sln`	Solution file
`dotnet new console`	Console application
`dotnet new classlib`	Class library
`dotnet new mvc`	ASP.NET Core Web App (Model-View-Controller)
`dotnet new xunit`	xUnit test project
`dotnet new -l`	Obtain a list of the available templates

dotnet sln

`dotnet sln list`	List all projects in a solution file
`dotnet sln todo.sln add todo-app/todo-app.csproj`	Add a C# project to a solution
`dotnet sln todo.sln remove todo-app/todo-app.csproj`	Remove a C# project from a solution
`dotnet sln todo.sln add **/*.csproj`	Add multiple C# projects to a solution using a globbing pattern

dotnet add

`dotnet add package Newtonsoft.Json`	Add Newtonsoft.Json NuGet package to a project
`dotnet add reference lib1/lib1.csproj lib2/lib2.csproj`	Add multiple project references to the project in the current directory
`dotnet add app/app.csproj reference **/*.csproj`	Add multiple project references using a globbing pattern on Linux/Unix

dotnet build

`dotnet build`	Build a project and all of its dependencies
`dotnet build --configuration Release`	Build a project and its dependencies using Release configuration
`dotnet build --runtime ubuntu.16.04-x64`	Build a project and its dependencies for a specific runtime (in this example, Ubuntu 16.04)

Starting with .NET Core 2.0, you don't have to run `dotnet restore` because it's run implicitly.

dotnet run

`dotnet run`	Run the project in the current directory
`dotnet run --project ./projects/proj1/proj1.csproj`	Run the specified project
`dotnet myapp.dll`	Run a framework-dependent app named `myapp.dll`

dotnet clean

`dotnet clean`	Clean the output of a project
`dotnet clean --configuration Release`	Clean a project built using the Release configuration

Only the outputs created during the build are cleaned. Both intermediate (obj) and final output (bin) folders are cleaned.

dotnet publish

`dotnet publish`	Publish the project in the current directory
`dotnet publish ~/projects/app1/app1.csproj`	Publish the application using the specified project file

The `dotnet publish` command's output is ready for deployment to a hosting system (for example, a server, PC, Mac, laptop) for execution.

dotnet ef

`dotnet ef migrations add`	Add a new migration
`dotnet ef migrations list`	List available migrations
`dotnet ef migrations remove`	Remove the last migration
`dotnet ef migrations script`	Generate a SQL script from migrations
`dotnet ef database update`	Update the database to a specified migration
`dotnet ef database drop`	Drop the database
`dotnet ef dbcontext list`	List available DbContext types
`dotnet ef dbcontext info`	Get information about a DbContext type
`dotnet ef dbcontext scaffold`	Scaffolds a DbContext and entity types for a database

By **Olcay Bayram** (oba)
cheatography.com/oba/
en.otomatikmuhendis.com

Published 30th June, 2018.
Last updated 30th June, 2018.
Page 1 of 2.

Sponsored by **ApolloPad.com**
Everyone has a novel in them. Finish Yours!
https://apollopad.com