

Getting Started with .NET Core 2.0 on macOS (and Windows, too)

Jeremy Clark
www.jeremybytes.com
@jeremybytes

What is .NET Core?

- Cross-Platform .NET
- Runs on macOS, Linux, and Windows
- Can be compiled to machine code based on platform/
architecture

What I like about .NET Core

- .NET Core 2.0 is **EASY** to get installed and configured.
- Self-hosted web applications with Kestrel.
- Simplified ASP.NET templates.
- Built-in Dependency Injection works for many scenarios.

What I like about .NET Core

- Command-line interface (CLI)
- Command-line help is actually helpful.
- The same tool chain can be used across platforms (CLI, Visual Studio Code).
- Cross-platform actually works!

Stuff you need to build applications with .NET Core 2.0

- .NET Core 2.x SDK
<https://www.microsoft.com/net/download/macos>
- Visual Studio Code (or Visual Studio for Mac)
<https://code.visualstudio.com/download>
- C# Extension for Visual Studio Code
<https://marketplace.visualstudio.com/items?itemName=ms-vscode.csharp>

CLI

Command-Line Interface

- dotnet
- dotnet new
- dotnet add package
- dotnet build
- dotnet run

dotnet -h

```
[Jeremys-MacBook-Pro:Core jeremy$ dotnet -h  
.NET Command Line Tools (2.0.0)  
Usage: dotnet [runtime-options] [path-to-application]  
Usage: dotnet [sdk-options] [command] [arguments] [command-options]
```

dotnet -h

SDK commands:

new	Initialize .NET projects.
restore	Restore dependencies specified in the .NET project.
run	Compiles and immediately executes a .NET project.
build	Builds a .NET project.
publish	Publishes a .NET project for deployment (including the runtime).
test	Runs unit tests using the test runner specified in the project.
pack	Creates a NuGet package.
migrate	Migrates a project.json based project to a msbuild based project.
clean	Clean build output(s).
sln	Modify solution (SLN) files.
add	Add reference to the project.
remove	Remove reference from the project.
list	List reference in the project.
nuget	Provides additional NuGet commands.
msbuild	Runs Microsoft Build Engine (MSBuild).
vstest	Runs Microsoft Test Execution Command Line Tool.

dotnet new

Templates	Short Name	Language	Tags
Console Application	console	[C#], F#, VB	Common/Console
Class library	classlib	[C#], F#, VB	Common/Library
Unit Test Project	mstest	[C#], F#, VB	Test/MSTest
xUnit Test Project	xunit	[C#], F#, VB	Test/xUnit
ASP.NET Core Empty	web	[C#], F#	Web/Empty
ASP.NET Core Web App (Model-View-Controller)	mvc	[C#], F#	Web/MVC
ASP.NET Core Web App	razor	[C#]	Web/MVC/Razor Pages
ASP.NET Core with Aurelia	aurelia	[C#]	Web/MVC/SPA
ASP.NET Core with Knockout.js	knockout	[C#]	Web/MVC/SPA
ASP.NET Core with Vue.js	vue	[C#]	Web/MVC/SPA
ASP.NET Core with Angular	angular	[C#]	Web/MVC/SPA
ASP.NET Core with React.js	react	[C#]	Web/MVC/SPA
ASP.NET Core with React.js and Redux	reactredux	[C#]	Web/MVC/SPA
ASP.NET Core Web API	webapi	[C#], F#	Web/WebAPI
global.json file	globaljson		Config
Nuget Config	nugetconfig		Config
Web Config	webconfig		Config
Solution File	sln		Solution
Razor Page	page		Web/ASP.NET
MVC ViewImports	viewimports		Web/ASP.NET
MVC ViewStart	viewstart		Web/ASP.NET

Creating & Running

- New Projects
 - `dotnet new webapi`
 - `dotnet new console`
- Build / Run
 - `dotnet build`
 - `dotnet run`

Adding Packages / References

- NuGet Packages
 - `dotnet add package Newtonsoft.Json`
- Add Reference
 - `dotnet add reference ../folder/my-library.csproj`

Solutions

- Solution Files
 - `dotnet new sln`
 - `dotnet sln add ./folder1/my-console.csproj`
 - `dotnet sln add ./folder2/my-library.csproj`

Async

- “async” Main in a console application
 - Open .csproj
 - Add “<LangVersion>latest</LangVersion>” to PropertyGroup section

ASP.NET Dependency Injection

- Auto-inject dependencies as parameters in Controller constructors.
- In Startup.cs
 - `services.AddSingleton<IPeopleProvider, StaticPeopleProvider>()`
- Lifetimes: Singleton, Scoped, Transient

Thank You!

- Jeremy Clark
 - jeremy@jeremybytes.com
 - [@jeremybytes](#)
 - <http://www.jeremybytes.com>