

.NET

*Free. Cross-platform. Open source.
A developer platform for building all your apps.*

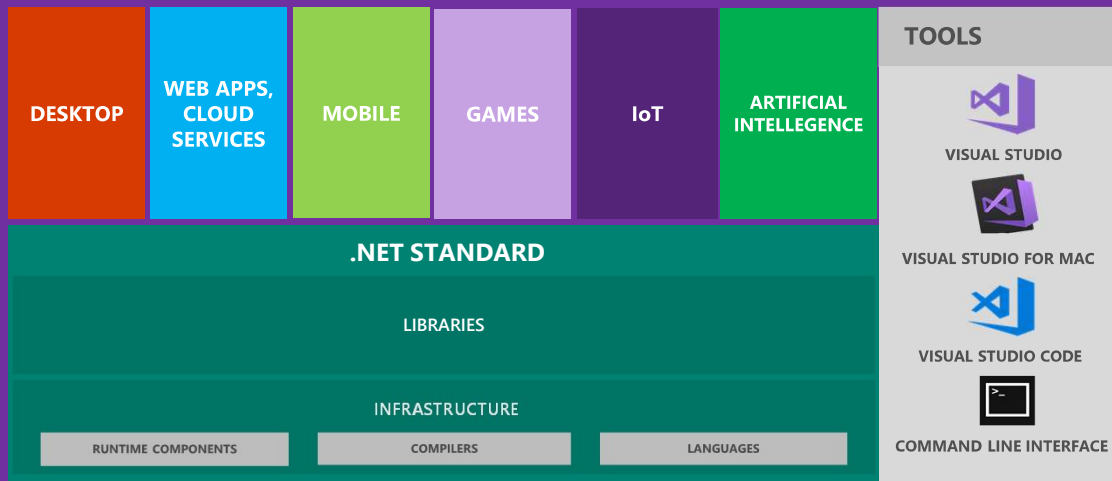
www.dotnet.net



What is .NET?

So what is .NET?

.NET is a software development platform



You can build anything with .NET. It is an entire software development platform that takes care of a lot of the heavy lifting for you when you want to build an application. Applications frameworks help you build the specific types of apps or workloads and enable you to literally build any app for any platform with any operating system. Each .NET workload shares a common infrastructure and .NET Standard library. This means not only are your .NET skills portable, but your *actual code* is portable no matter what you're building. This makes it easy to share reusable components (called libraries) across the breadth of applications people build.

Additionally, there are a broad set of development tools that makes it really productive to write, debug, build and manage code bases.

See: www.dot.net

You can write code with many .NET languages

C# (c-sharp)

- C# is a simple, modern, object-oriented, and type-safe programming language
- Its roots in the C family of languages makes C# immediately familiar to C, C++, Java, and JavaScript programmers

```
var names = new List<String>
{
    "Ana",
    "Felipe",
    "Emillia"
};

foreach (var name in names)
{
    Console.WriteLine($"Hello {name}");
}
```

F# (f-sharp)

- F# is a cross-platform, open-source, functional programming language for .NET
- It also includes object-oriented and imperative programming

```
let numbers = [ 1; 2; 3; 4; 5; 6; 7; 8; 9; 10 ]

let square x = x * x
let isodd x = x % 2 <> 0

let squaresOfOdds =
    numbers
    |> List.filter isOdd
    |> List.map square

printfn "%A" squaresOfOdds
```

Visual Basic

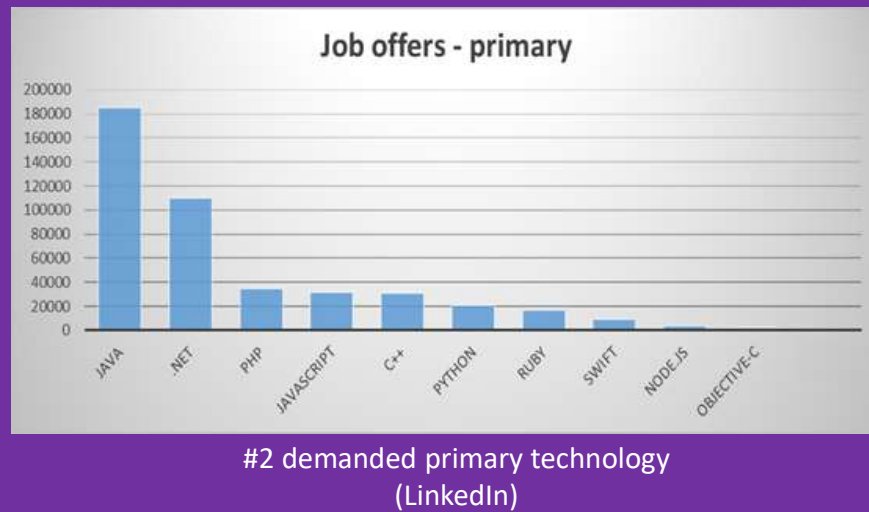
- Visual Basic is an approachable language with a simple syntax for building type-safe, object-oriented apps

```
Dim names As New List(Of String)({
    "Ana",
    "Felipe",
    "Emillia"
})

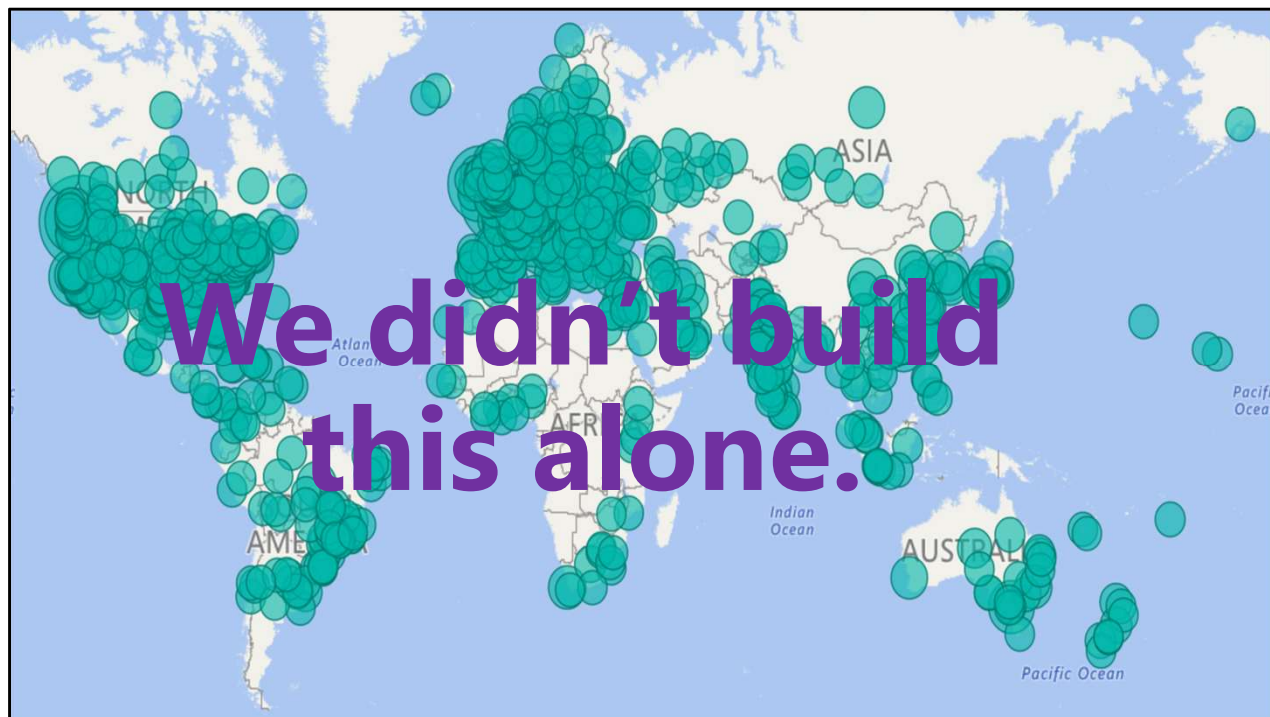
For Each name In names
    Console.WriteLine($"Hello {name}")
Next
```

(Speak to slide bullet points.)

.NET is a sought after job skill

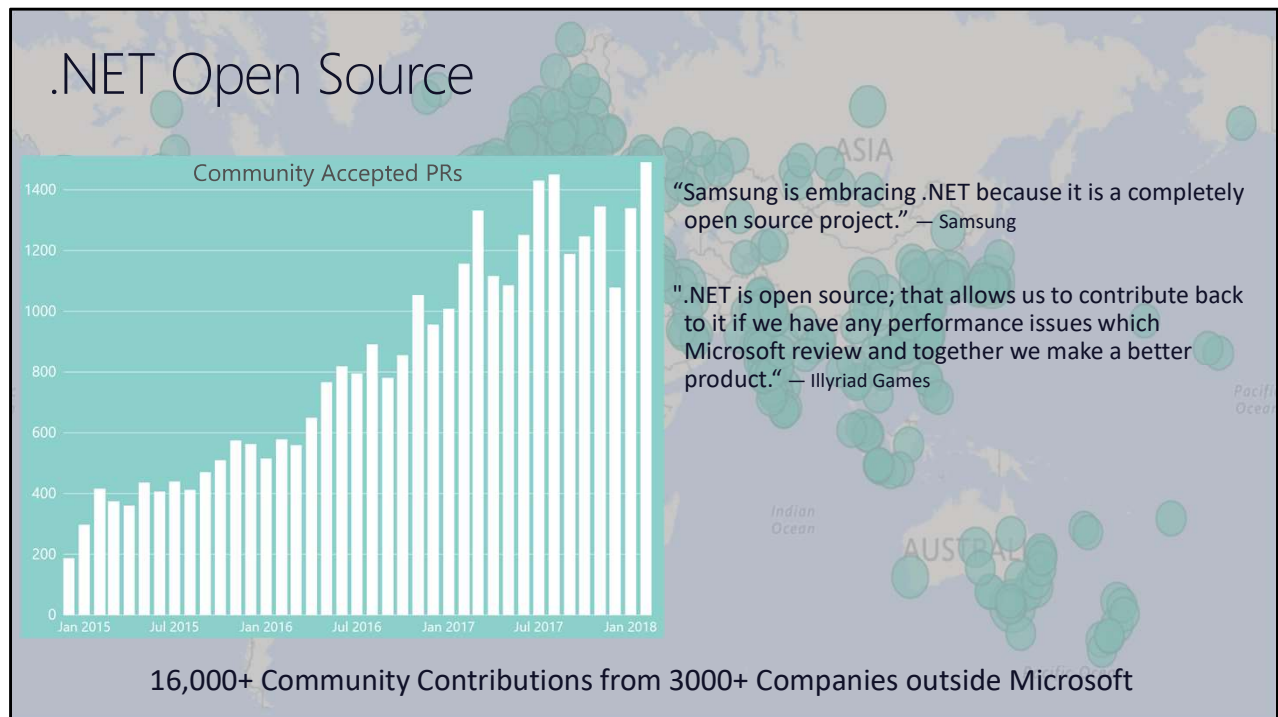


Everybody wants to have a successful career and that means finding a job. There are a LOT of .NET jobs available since it's been around for over 15 years. And in general, there are a huge amount of programming jobs.



Of course, we couldn't do any of this without the open source community.

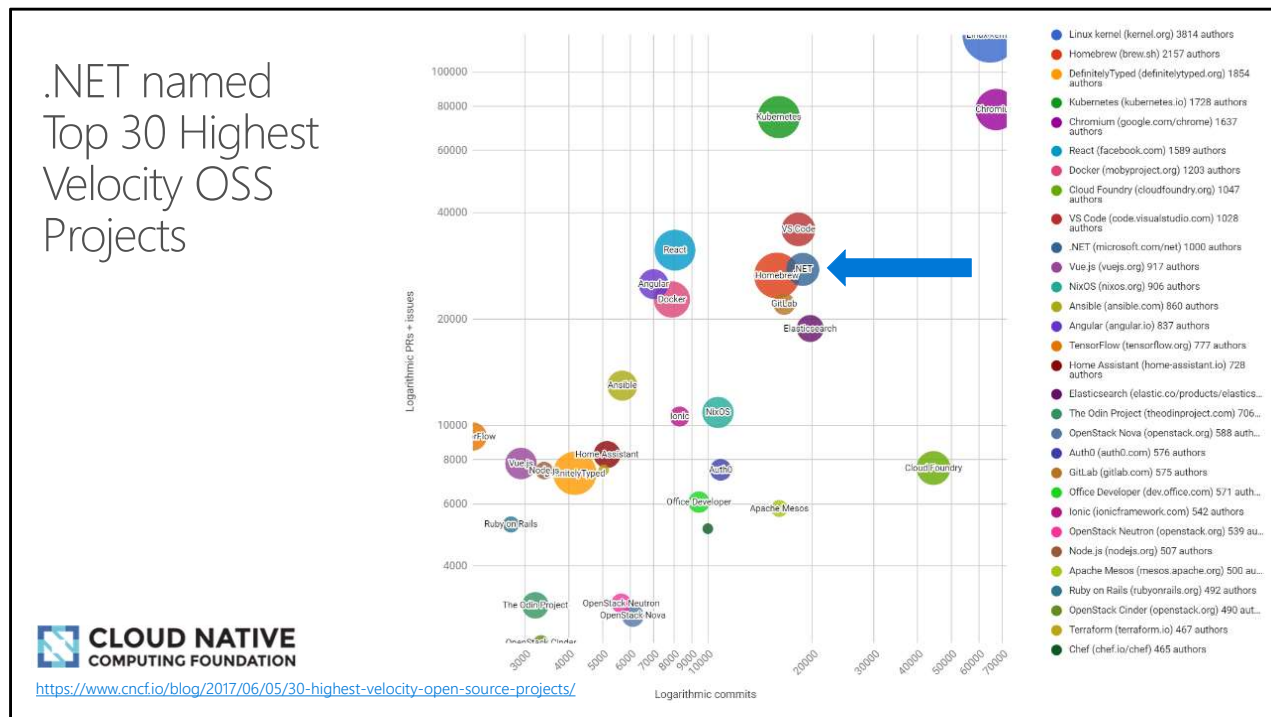
Open source is about accepting people's diverse opinions and combined intelligence to build something amazing. There are thousands of people all over the world contributing to the .NET platform.



We've seen companies taking a bet on .NET because it's a completely open source project. More than 16,000 contributors* from 3300 companies have contributed to .NET Core and related open source repositories, with over half of the contributions coming from outside Microsoft.

Samsung is a big partner and are releasing their new Smart TVs that support .NET. They plan to expand to all their IoT devices too. Illyriad Games, specifically Ben Adams, contributed many performance enhancements to the platform. They build a MMO with hundreds of thousands of concurrent users so performance is super important to them. These enhancements not only benefit them but everyone that uses .NET.

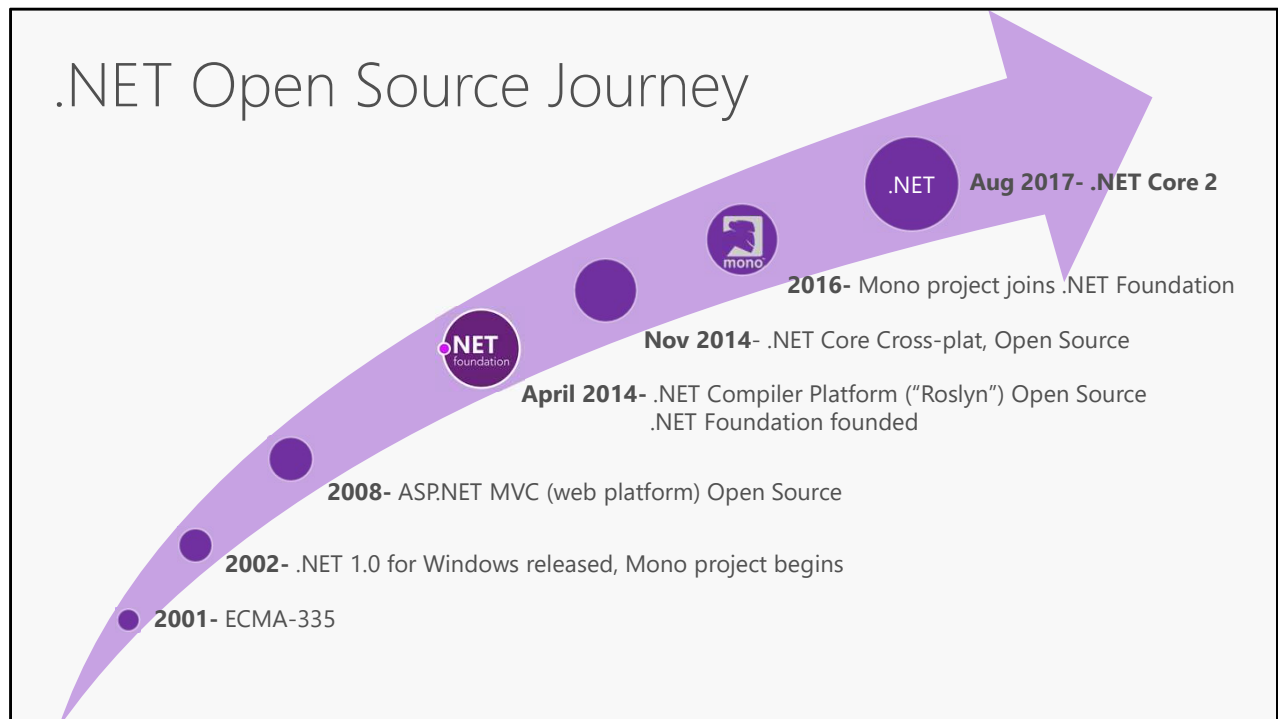
*contributor = Submitted a Pull Request or *Created* an Issue



Jim Zemlin, Executive director of Linux foundation said “There are 10’s of millions of open source projects, invest in the ones with sustainable ecosystems”. .NET is one of those projects. June 5th 2017, CNCF released an analysis of the top 30 highest velocity open source projects today. This is a logarithmic scale to measure OSS project velocity with commits on the X axis and PRs and issues on the Y axis, with # of authors indicated by the size of the circle. The farther upper right the more activity a project has. Linux Kernel is #1, followed by Chromium, Kubernetes and VSCode. Notice .NET is the #1 application framework.

Updated live:

https://cdn.rawgit.com/cncf/velocity/d6f852d9/charts/top_30_bubble_chart_2017.html



This didn't happen overnight. Here's a little history of .NET open source.....

Dec 2001-Feb 2002.

A new platform is born. Along with HP, Intel and others, the ECMA-335 standard was created that defined a common language infrastructure to support multiple programming languages. C# and Visual Basic.NET were released and F# came later in 2007, but there are over 20 other .NET languages today. Visual Studio .NET was released and included C#, VB, C++ development all in one box. This was the first IDE that was truly integrated across multiple languages.

Mono project begins. The CLI spec gave others the ability to create their own .NET implementations. Even though Microsoft built the first .NET Framework for Windows-only, the spec was intentionally portable across OSes and chipsets. The Mono project began, spearheaded by

Miguel de Icaza, with the goal to implement Microsoft's new .NET development platform on Linux and Unix-like platforms. Later, Miguel started Xamarin which focused on cross-platform, native, mobile development with C#, built upon Mono. This allows developers to use C# and .NET to build apps for iOS and Android. Unity games development also emerged from Mono.

2008.

ASP.NET MVC web development stack is released to CodePlex as open source. This was the first application development framework from Microsoft to be released as open source. The underlying runtime and compilers were still closed however.

2014.

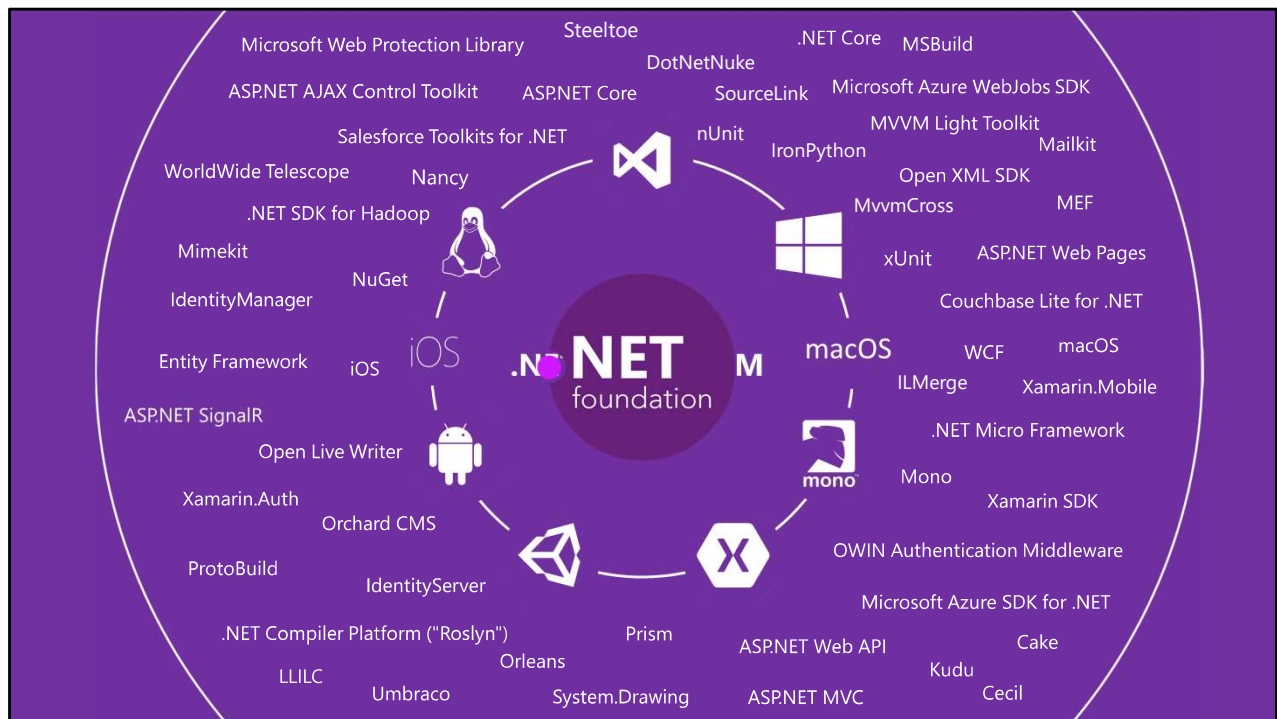
Hell freezes over & pigs fly. Early 2014 at Microsoft's BUILD conference, Anders Hejlsberg, the father of C#, releases the .NET Compiler Platform "Roslyn" as open source on stage. Later in November, .NET Core project begins in the open. The technology world is shocked, and the .NET community is excited. .NET Core is a new cloud-native implementation of .NET that is geared for cross-platform, hyper-scale services as well as small IoT devices. It's meant to bring .NET into the next 15 years of computing. And the community has been extremely supportive.

2016.

Mono comes home. In early 2016, Microsoft finally acquires Xamarin and brings Miguel de Icaza into Developer Division. Mono joins the .NET foundation and is officially supported and contributed to by Microsoft. The Microsoft community officially meets the Mono community.

2017.

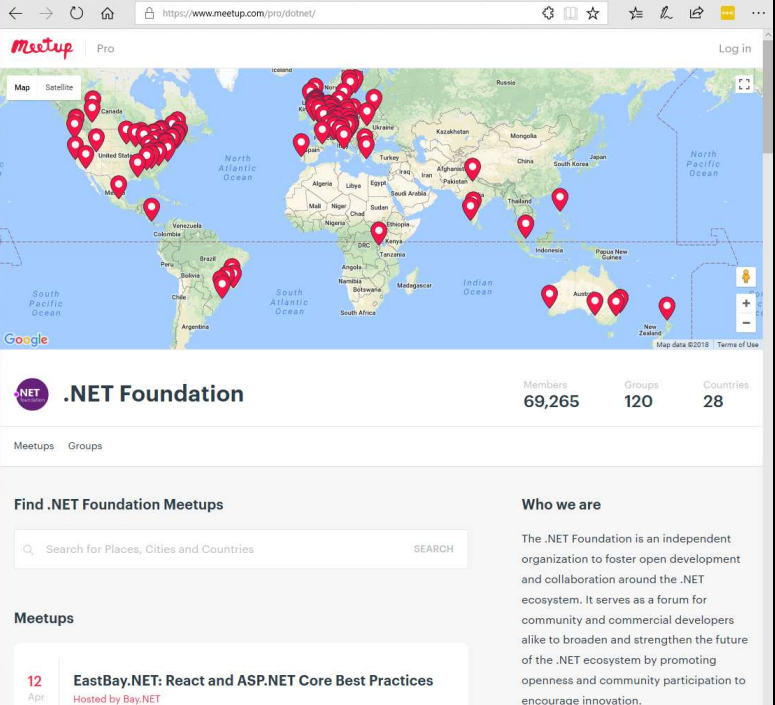
.NET Core 2.0 Released. Our cross platform and open source implementation of .NET finally releases to the world with unified tooling support across multiple operating systems and editors.



The .NET Foundation is our center of gravity for open development and collaboration around the .NET ecosystem. The .NET Foundation has over 60 projects and hundreds of repos under its stewardship. Open Source Software foundations provide protection, support, services and best practices for helping each project be successful and to grow the ecosystem of people and software.

Meetups

- These User Groups have presentations on all sorts of .NET topics
- 120 groups and counting worldwide, just on meetup.com
- A lot of UGs aren't on meetup.com










The screenshot displays the Meetup.com interface for the .NET Foundation group. At the top, there's a navigation bar with the Meetup logo and a 'Pro' badge. Below this is a world map with numerous red location pins, indicating the global reach of .NET meetups. To the right of the map, there's a 'Log in' button. Below the map, a summary section for the '.NET Foundation' group shows 69,265 members, 120 groups, and 28 countries. A search bar is provided to find local meetups. Below the search bar, there's a section titled 'Meetups' with a list of featured events, including 'EastBay.NET: React and ASP.NET Core Best Practices'.

We're a huge community and the .NET Foundation is key to help foster it. We have over 4 million developers worldwide and .NET has grown its developer base by 1 million over the last year. There are meetups all over the world that speak on all sorts of .NET topics and application development. Here are the ones on meetup.com.



And it's not just individual developers, it's large companies that are invested in the future of .NET. .NET is bigger than Microsoft. The technical steering group of the .NET Foundation is responsible for guiding .NET into the future together with industry partners.



[RayGun](#)
"Using the same-size server, we were able to go from 1,000 requests per second per node with Node.js to 20,000 requests per second with .NET Core."
- John-Daniel Trask, CEO and Co-Founder

[Jet.com Inc.](#)
"The event-driven, microservices paradigm eliminated a lot of the overhead that comes with a service-oriented architecture."
- Mike Hanrahan, CTO

[Age of Ascent, Illyriad Games](#)
"Our game microservices are built using ASP.NET Core which gives us superior performance. ASP.NET is open source, that allows us to contribute back to it if we have any performance issues which Microsoft review and together we make a better product."
- Ben Adams, CTO and Co-founder

[United Parcel Service \(UPS\)](#)
"Xamarin allowed us to develop a single code base in C# and deploy the application to two completely different mobile phone ecosystems."
- Scott Hildebrand, Application Development Manager

[AstroReality, Quantum Technologies](#)
"We think .NET Core is mature, very powerful, and a great choice for advanced modern apps."
- Zerlot Ma, Co-founder and CEO

[Tencent](#)
"The cross-platform capabilities of .NET Core were very important to us. Also, Microsoft designed .NET Core with a microservice architecture in mind, and that fits right in with our redesign plans."
- Shanyou Zhang: Senior Software Architect

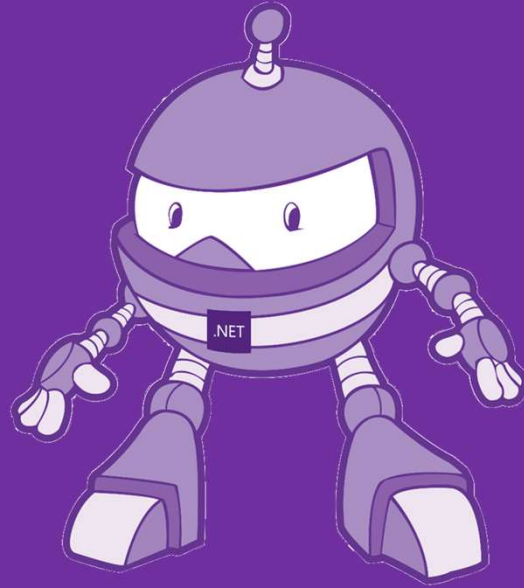
[NetEase](#)
".NET Core doesn't just help us get to market faster, it also enables our developers to focus on the core logic and what really matters, our players' experience. They don't have to deal with the communication gap and different ways of implementing the algorithm on the back and front end. They can just work on finding better ways to implement their ideas."
- Feng Zhou, Product Director

[GoDaddy](#)
"Services can be developed more quickly, perform faster in production, and scale better if they're written using .NET Core with C#."
- Jeremiah Gowdy: Principal Software Architect

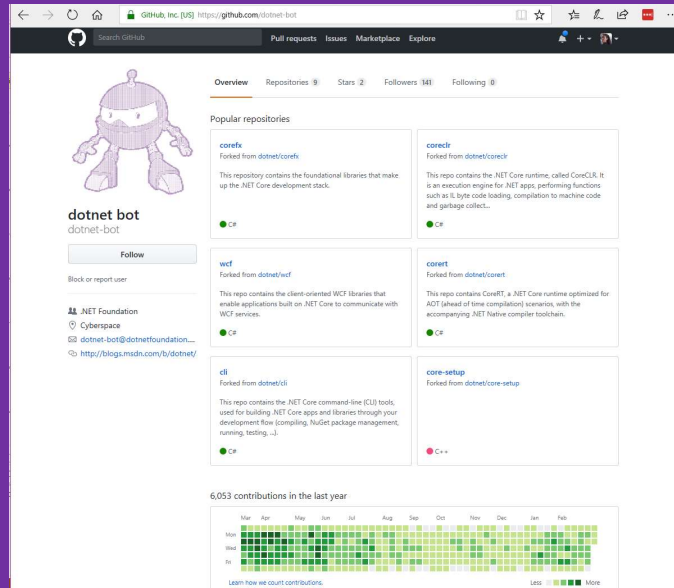
And many more: microsoft.com/net/customers

Our customers have been embracing the productivity in our platform, languages and tools for 15+ years. We're now gaining existing customers as well as new ones on our cloud-native, cross-platform, and device-native approaches with these new architectures. Here are just a few. There are many more (and their stories) on our website.

This is our mascot, dotnet-bot

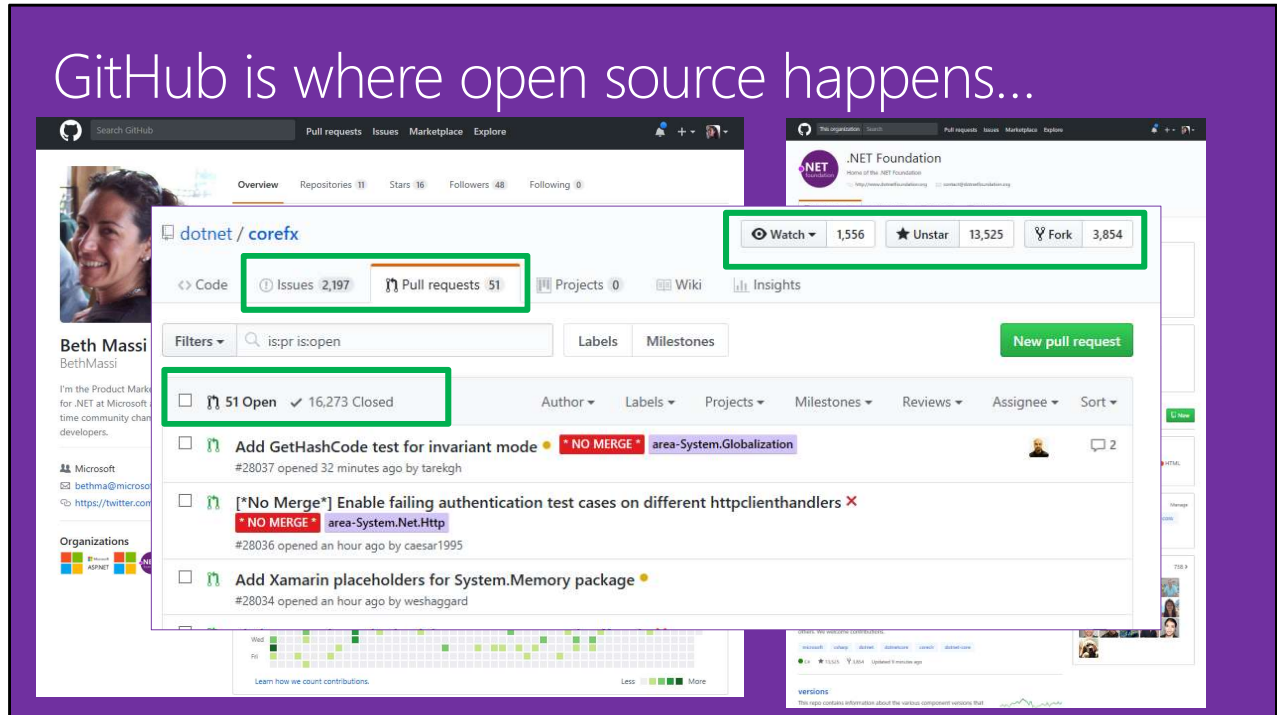


You can find dotnet-bot on GitHub



Dotnet-bot helps us check the contributions (called Pull Requests) to make sure they are the developer's original work. Dotnet-bot does this by checking if people have signed a contributor license agreement. This makes sure that the code that is submitted isn't stolen from another product or company.

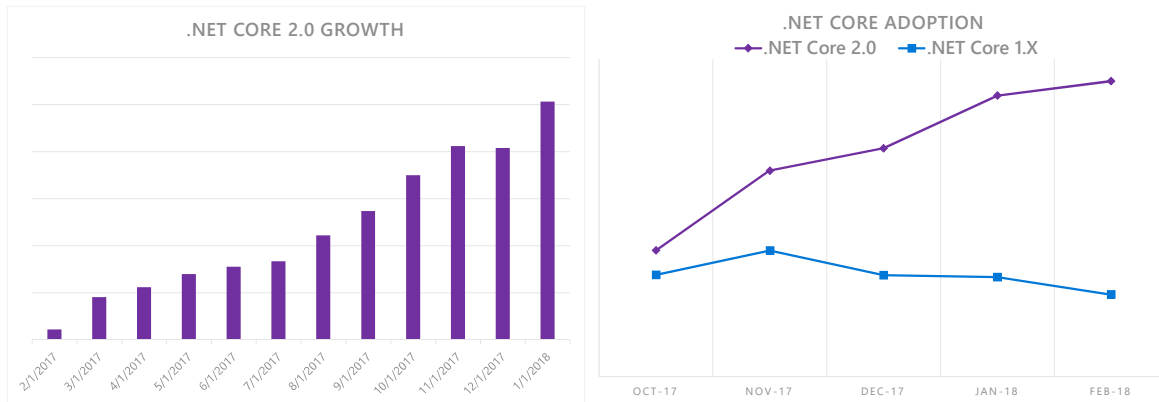
GitHub is where open source happens...



I encourage you to look through GitHub – there are millions of open source projects there, some big, some small. To give you a sense of the size of .NET, take a look at how many PRs we've had, open discussions happening in issues, and stars (or likes) we've got.

.NET Core Growth

Over Half Million Active* .NET Core 2.0 Developers!



* Active = unique monthly developers with 2+ days of development per month.

And our usage is growing. As of February 2018, if we take a look at just the cross-platform, web workload (.NET Core) you can see we are already over a half million active developers per month. Active means unique developers with 2+ days of development per month.

On average we're seeing double digit percentage growth of .NET Core MoM since 2.0 release in August. Metric includes .NET Core CLI + .NET Core in Visual Studio, 2day+ Active.

.NET all up across other workloads is also growing rapidly. We have 4M active developers. We get about 450K new developers/Month and we see a 15% growth in Community PR's MoM.

Learn how to program with C#

www.dot.net

In-Browser Tutorial

Microsoft .NET Downloads Learn Architecture Docs Customers Community Support

Free. Cross-platform. Open source. A developer platform for building all your apps.

Get Started Download

Supported on Windows, Linux, and macOS

Web Build web apps and services for Windows, Linux, macOS, and Docker.

Mobile Use a single codebase to build native mobile apps for iOS, Android, and Windows.

Desktop Create beautiful and compelling desktop apps for Windows and macOS.

Gaming Develop 2D and 3D games for the most popular desktops, phones, and consoles.

Machine Learning & AI Add vision algorithms, speech processing, predictive models, and more to your apps.

Internet of Things Make IoT apps with native support for the Raspberry Pi and other single-board computers.

```
1 using System;
2 using System.Collections.Generic;
3 using System.Linq;
4
5 namespace MyApp
6 {
7     class Program
8     {
9         static void Main()
10         {
11             Console.WriteLine("Hello Beth!");
12             Console
13         }
14     }
15 }
```

Run Code Next Step Hey Beth! Let's keep coding.

CONSOLE
Hello Beth!

So you want to learn .NET? Start with a language. We make it easy to get started learning right in your browser.

Demo

Get started with .NET in your browser

In-Browser tutorial:

www.dot.net → Get Started → Select In-browser tutorial

<https://www.microsoft.com/net/learn/in-browser-tutorial/1>

-- Have the class walk through some of the lessons.

Make it easier with some free tools

www.VisualStudio.com



Visual Studio

A fully-featured, integrated development environment (IDE) for developing .NET apps on a Windows PC development machine.



Visual Studio Code

Open source, cross-platform editor with .NET support.



Visual Studio for Mac

A fully-featured editor for developing .NET apps on a Mac OS development machine.

We also have free tools that can help you build your apps.

Demo

Get started with .NET on your computer

Machine demo:

Setup – install dotnet SDK

->> . Got to www.dot.net/core – walk through the setup instructions for your operating system

->> . [Optional] Install an editor like Visual Studio Code (any OS) or Visual Studio (Windows) from www.visualstudio.com

Open terminal/command prompt

>dotnet new

Show all the templates you can use

>dotnet new web

Creates a web application

>dotnet run

Starts web app. Navigate a browser to localhost:5000 to show “Hello World”

> CTRL+C to stop web server

>code .

Open VSCode

Run through this tutorial:

<https://docs.microsoft.com/en-us/dotnet/core/tutorials/with-visual-studio-code>

