DotNet
2021

ONLINE TECH CONFERENCE

#DotNet2021

22nd June 2021

Minimal APIs
.NET 6

www.dotnet2021.com

Minimal APIs

APIs without controllers

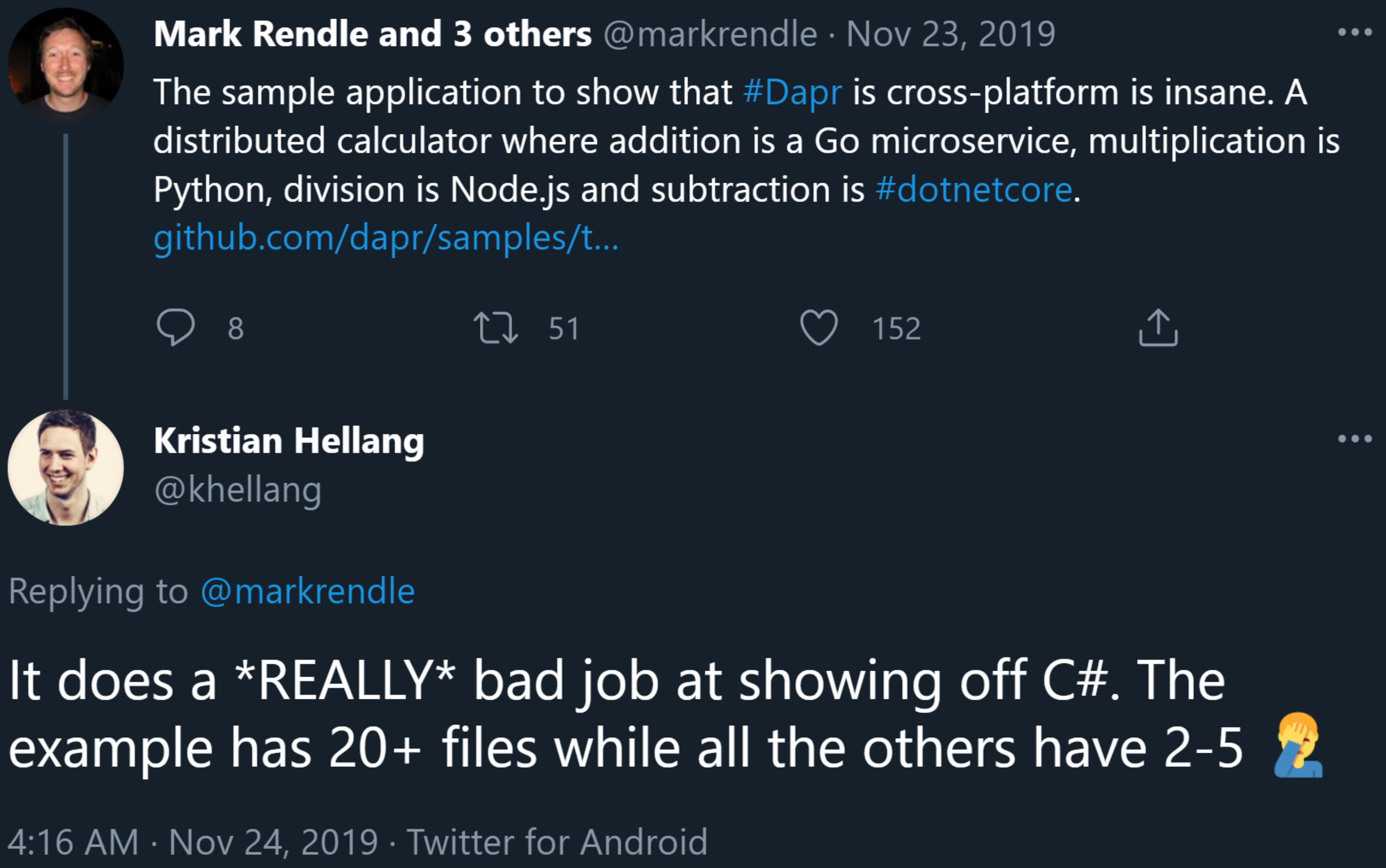
The Team

Stephen Halter
Senior Software Engineer

Maria Naggaga
Senior Program Manager

David Fowler
Partner Software Architect

The Tweet



Mark Rendle and 3 others @markrendle · Nov 23, 2019

The sample application to show that #Dapr is cross-platform is insane. A distributed calculator where addition is a Go microservice, multiplication is Python, division is Node.js and subtraction is #dotnetcore. github.com/dapr/samples/t...

💬 8 🔁 51 ♡ 152

Kristian Hellang
@khellang

Replying to @markrendle

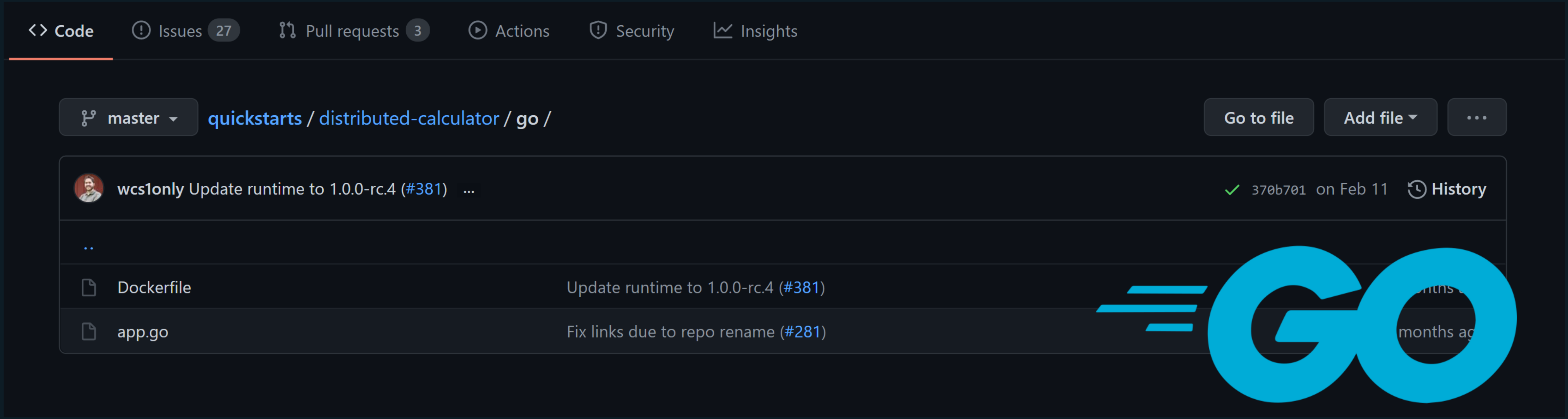
It does a *REALLY* bad job at showing off C#. The example has 20+ files while all the others have 2-5 🤦‍♂️

4:16 AM · Nov 24, 2019 · Twitter for Android

It’s true though

Looking around the room

Gif : Schitts Creek CBC

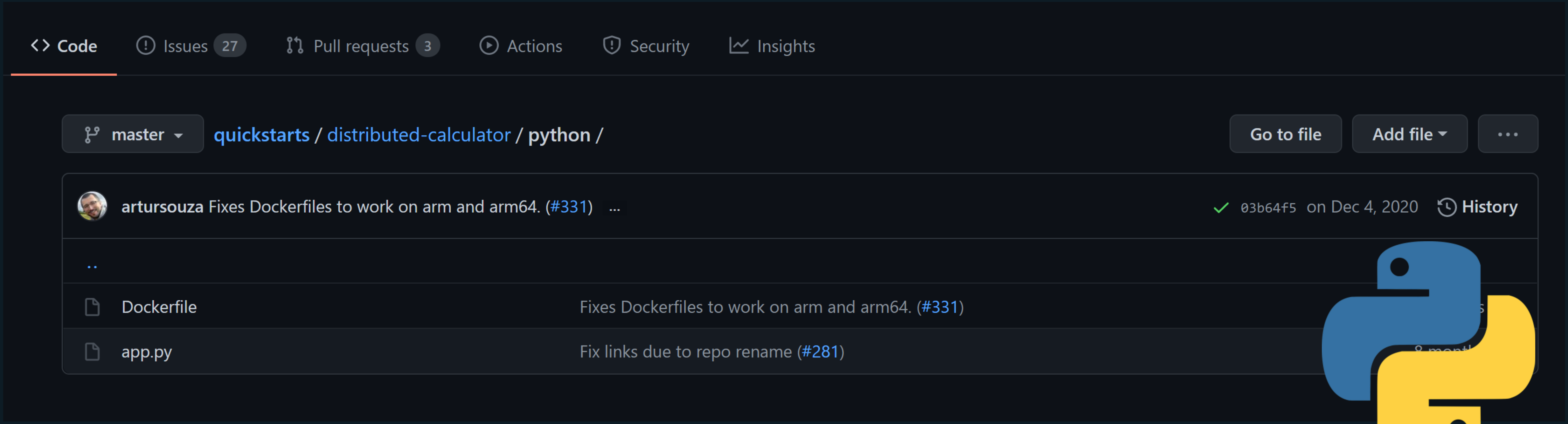


quickstarts / distributed-calculator / go /

wcs1only Update runtime to 1.0.0-rc.4 (#381) … ✓ 370b701 on Feb 11

..		
Dockerfile	Update runtime to 1.0.0-rc.4 (#381)	nths
app.go	Fix links due to repo rename (#281)	months ag

dapr



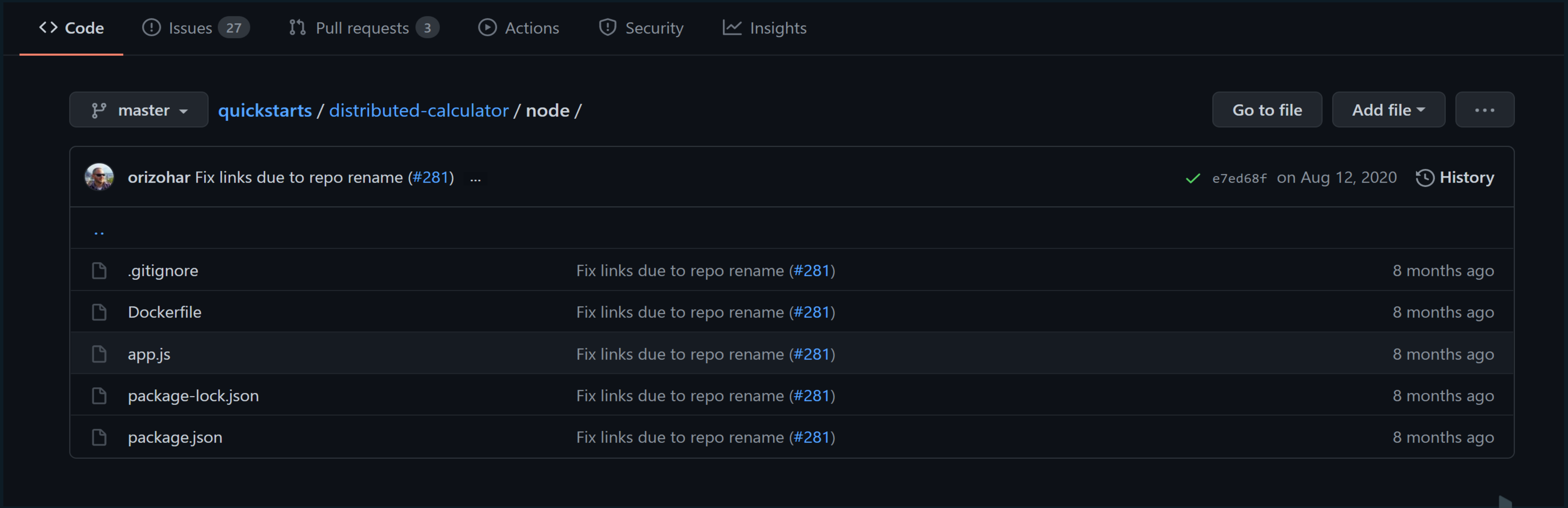
quickstarts / distributed-calculator / python /

artursouza Fixes Dockerfiles to work on arm and arm64. (#331) … 03b64f5 on Dec 4, 2020

..

Dockerfile — Fixes Dockerfiles to work on arm and arm64. (#331)

app.py — Fix links due to repo rename (#281)

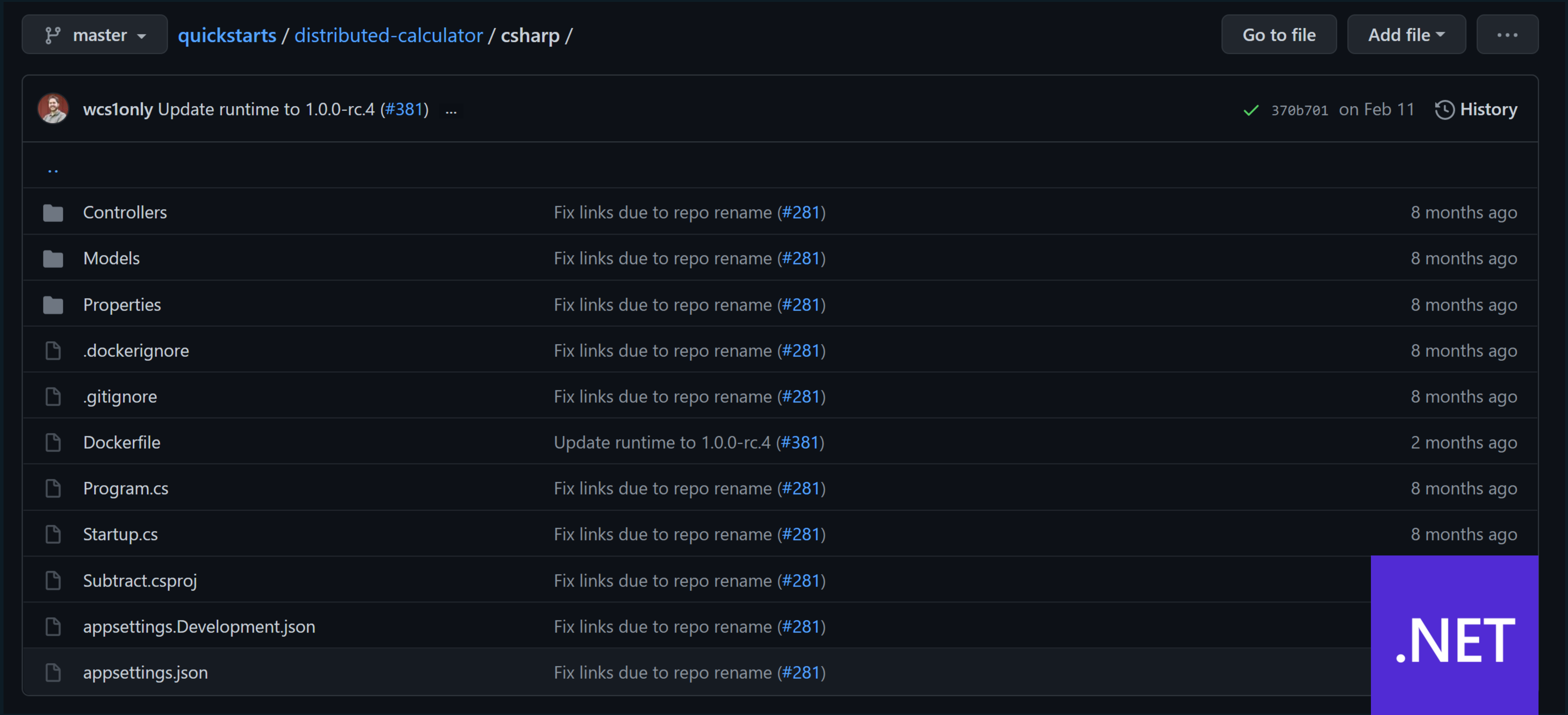


quickstarts / distributed-calculator / node /

orizohar Fix links due to repo rename (#281) … ✓ e7ed68f on Aug 12, 2020

..

.gitignore	Fix links due to repo rename (#281)	8 months ago
Dockerfile	Fix links due to repo rename (#281)	8 months ago
app.js	Fix links due to repo rename (#281)	8 months ago
package-lock.json	Fix links due to repo rename (#281)	8 months ago
package.json	Fix links due to repo rename (#281)	8 months ago



quickstarts / distributed-calculator / csharp /

wcs1only Update runtime to 1.0.0-rc.4 (#381) … ✓ 370b701 on Feb 11

..

Controllers	Fix links due to repo rename (#281)	8 months ago
Models	Fix links due to repo rename (#281)	8 months ago
Properties	Fix links due to repo rename (#281)	8 months ago
.dockerignore	Fix links due to repo rename (#281)	8 months ago
.gitignore	Fix links due to repo rename (#281)	8 months ago
Dockerfile	Update runtime to 1.0.0-rc.4 (#381)	2 months ago
Program.cs	Fix links due to repo rename (#281)	8 months ago
Startup.cs	Fix links due to repo rename (#281)	8 months ago
Subtract.csproj	Fix links due to repo rename (#281)	
appsettings.Development.json	Fix links due to repo rename (#281)	
appsettings.json	Fix links due to repo rename (#281)	

.NET Feels ....

Complex

Gif: Kim's Convenience CBC

.NET Feels ....

Complex

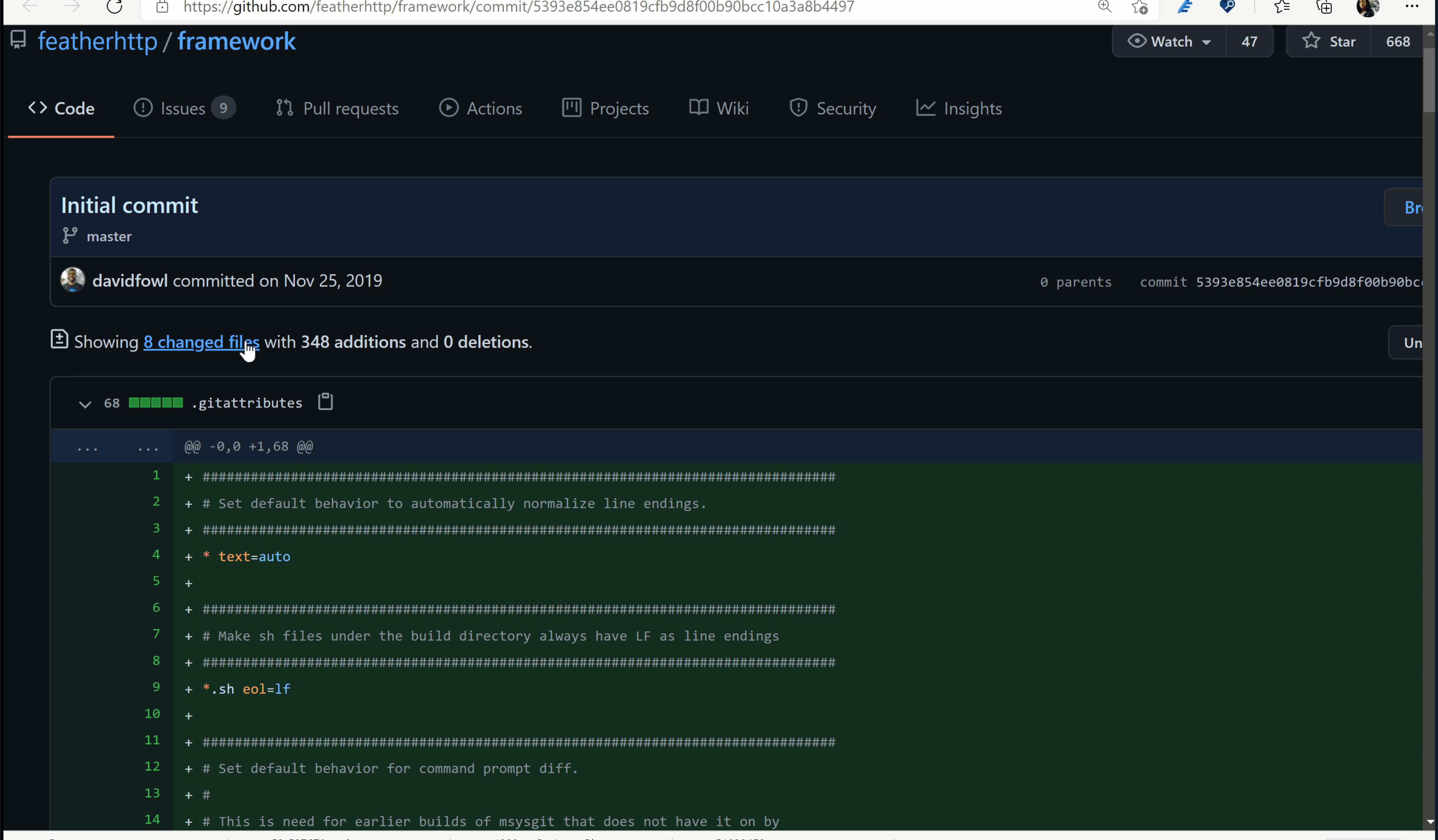
Gif: Kim's Convenience CBC

Ceremonial

Gif: Queen Elizabeth II 1953

Challenge Accepted

Hello World - Featherhttp



Initial commit

master

davidfowl committed on Nov 25, 2019

0 parents commit 5393e854ee0819cfb9d8f00b90bc

Showing 8 changed files with 348 additions and 0 deletions.

68 .gitattributes

```
@@ -0,0 +1,68 @@
+ ###############################################################################
+ # Set default behavior to automatically normalize line endings.
+ ###############################################################################
+ * text=auto
+
+ ###############################################################################
+ # Make sh files under the build directory always have LF as line endings
+ ###############################################################################
+ *.sh eol=lf
+
+ ###############################################################################
+ # Set default behavior for command prompt diff.
+ #
+ # This is need for earlier builds of msysgit that does not have it on by
```

Assembling a crew

D.F

Did you see twitter ? It started a spicy conversation on how complex .NET is in comparison to other frameworks.

M.N

I have been telling you all for ages….

D.F

Want to change it ?

M.N

I am in! We should get Charles to run the user study and customer development. He's work is EPIC!

Goals

Minimal APIs

- Reduce complexity for new people

Goals

Minimal APIs

- Reduce complexity for new people
- Reduce ceremony for the all developers

Goals

Minimal APIs

- Reduce complexity for new people
- Reduce ceremony for the all developers
- Embrace minimalism
  - Taking the lessons from .NET Interactive

Goals

Minimal APIs

- Reduce complexity for new people
- Reduce ceremony for the all developers
- Embrace minimalism
  - Taking the lessons from .NET Interactive
- We grow with you
  - Optimizing for a happy path

Conversations with new people

Outside our comfort zone

User Study | Format

Interviewed 16 non-.NET developers and invited the complete a tutorial.

Format

.NET Perceptions before → Tutorial → .NET Perceptions after

User Study

Perceptions Before

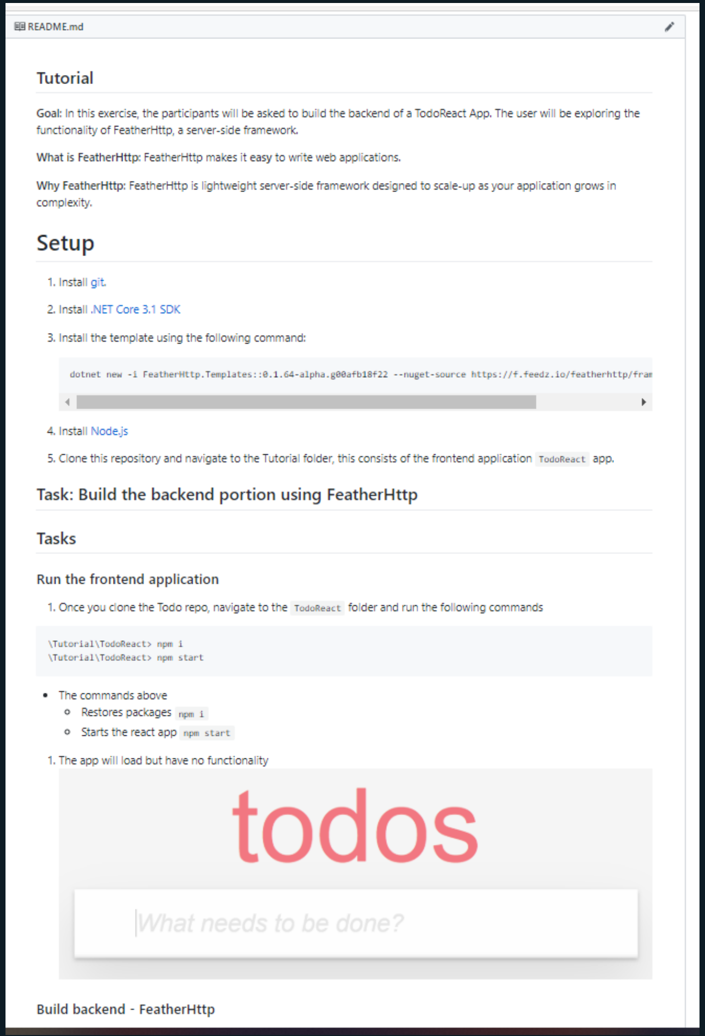
“When I think of .NET I think of early 2000 late 1990s era of web developer”
~ Fullstack JavaScript Dev at a mid size company

“I don't have a great perception. It's mostly enterprise and I have tried to stay away from that having coming from a military background.”
~ CTO at Startup Python Dev

“I have worked mostly in Startups it isn't widely adopted”
~ GO Developer at a Startup

User Study

Tutorial



README.md

Tutorial

Goal: In this exercise, the participants will be asked to build the backend of a TodoReact App. The user will be exploring the functionality of FeatherHttp, a server-side framework.

What is FeatherHttp: FeatherHttp makes it easy to write web applications.

Why FeatherHttp: FeatherHttp is lightweight server-side framework designed to scale-up as your application grows in complexity.

Setup

1. Install git.
2. Install .NET Core 3.1 SDK
3. Install the template using the following command:

```
dotnet new -i FeatherHttp.Templates::0.1.64-alpha.g00afb18f22 --nuget-source https://f.feedz.io/featherhttp/fram
```

4. Install Node.js
5. Clone this repository and navigate to the Tutorial folder, this consists of the frontend application `TodoReact` app.

Task: Build the backend portion using FeatherHttp

Tasks

Run the frontend application

1. Once you clone the Todo repo, navigate to the `TodoReact` folder and run the following commands

```
\Tutorial\TodoReact> npm i
\Tutorial\TodoReact> npm start
```

- The commands above
  - Restores packages `npm i`
  - Starts the react app `npm start`

1. The app will load but have no functionality

Build backend - FeatherHttp

Goal: Build the backend to Todo React App using .NET

User Study

Perceptions After

“It's totally different. I feel like C# has similar syntax to JavaScript so it feels like home”
~ Fullstack JavaScript Dev at a mid size company

“ My opinion of .NET might have been unfair…
It comes down to how many Python I can hire vs .NET. … it’s about public perception.”
~ CTO at Startup Python Dev

“When I think of .NET I think of something that is difficult to use. From this it doesn't seem that hard even though I am scratching the surface.”
~ GO Developer at a Startup

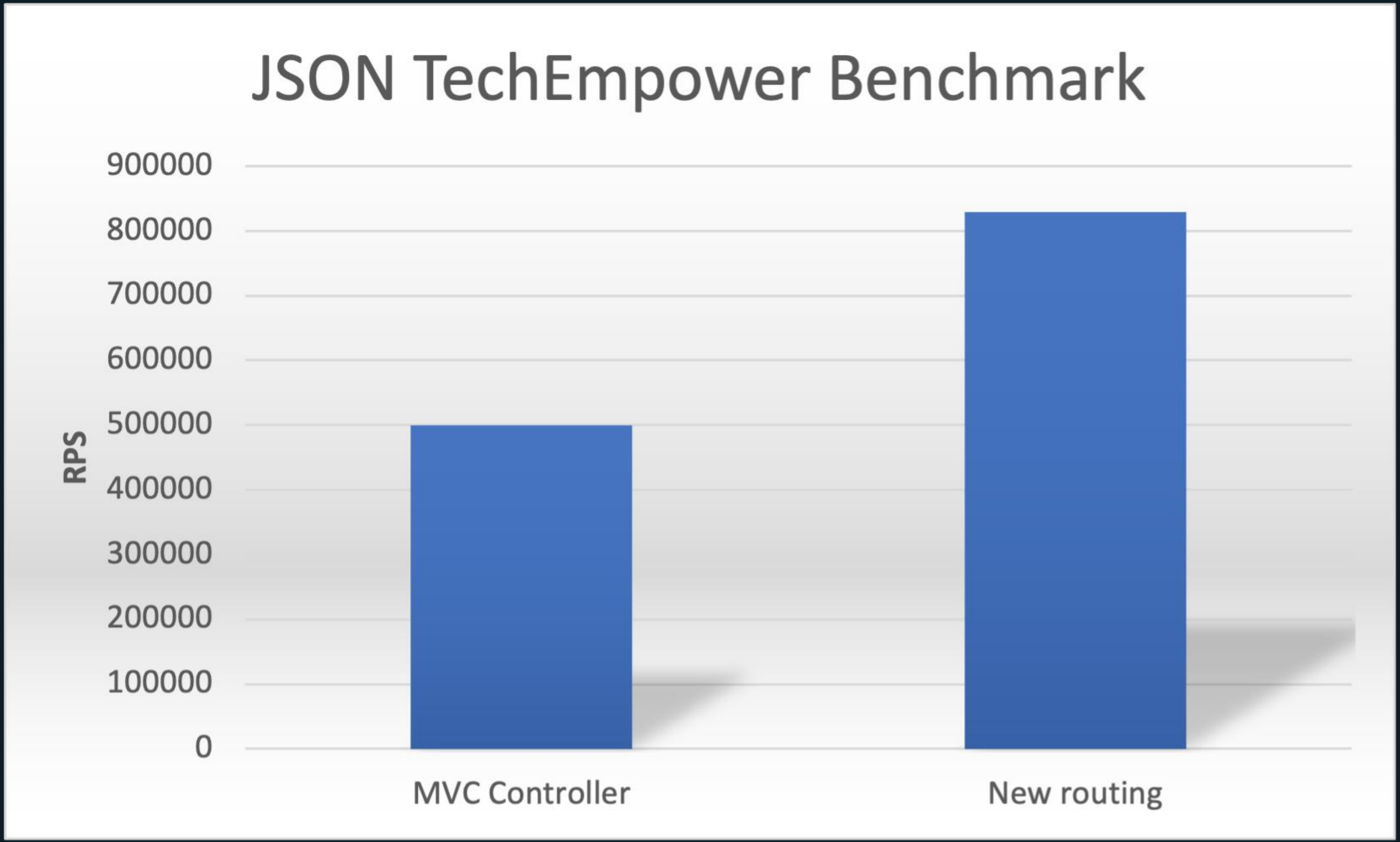
Allow me to reintroduce myself

Jay Z – PSA 2008

Demo: Minimal ASP.NET Core

Details | Where we are going

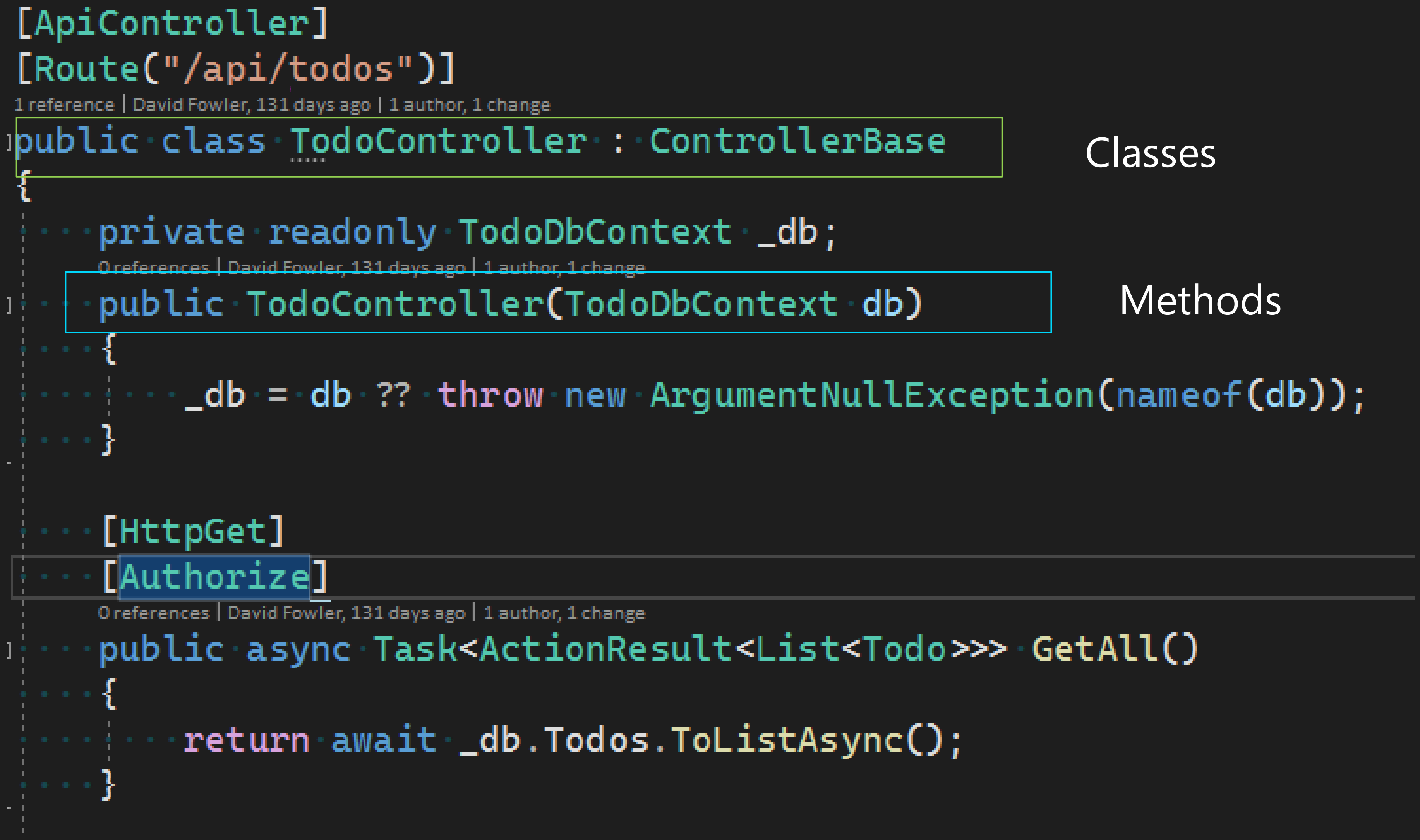
- High Performance



Details | Where we are going

- High Performance
- New C# Features : Improved Lambdas, Global Usings, Top Level Statements
- New primitives in ASP.NET Core
- Basic model binding
- Open APIs
- Shared Foundation : Use MVC features you love minus the need for a controller.

.NET APIs | Controller based APIs



```
[ApiController]
[Route("/api/todos")]
1 reference | David Fowler, 131 days ago | 1 author, 1 change
public class TodoController : ControllerBase
{
    private readonly TodoDbContext _db;
    0 references | David Fowler, 131 days ago | 1 author, 1 change
    public TodoController(TodoDbContext db)
    {
        _db = db ?? throw new ArgumentNullException(nameof(db));
    }

    [HttpGet]
    [Authorize]
    0 references | David Fowler, 131 days ago | 1 author, 1 change
    public async Task<ActionResult<List<Todo>>> GetAll()
    {
        return await _db.Todos.ToListAsync();
    }
```

Classes

Methods

.NET APIs | After without controllers

Attributes

Lambdas

```
app.MapGet("/api/todos", [Authorize] (TodoDbContext db) => db.Todos.ToListAsync());
```

.NET APIs | New primitives: Hosting model

```
public class WebApplicationBuilder
{
  IWebHostEnvironment Environment { get; }
  IServiceCollection Services { get; }
  Configuration Configuration { get; }
  ILoggerBuilder Logging { get; }

  IHostBuilder WebHost { get; }
  IHostBuilder Host { get; }
}

public class WebApplication : IHost, IApplicationBuilder, IEndpointRouteBuilder
{
  WebApplication Create(string[] args);
  WebApplicationBuilder CreateBuilder(string[] args);
  ...
}
```

.NET APIs | New primitives

```
public interface IResult
{
   Task ExecuteResultAsync(HttpContext context);
}
```

.NET APIs

New primitives

```
public static class RequestDelegateFactory
{
  RequestDelegate Create(Delegate action);
}
```

.NET APIs | New primitives

```
// New streamlined hosting model
var app = WebApplication.Create(args);

// New natural typed lambdas
var handler = async (int id) : IResult =>
{
   // Use result types from MVC
   return new JsonResult(...);
};

// Sprinkle magic sauce on it
var requestDelegate = RequestDelegateFactory.Create(handler);

// Run as middleware
app.Run(requestDelegate);
```

A nod
To the community

- NancyFx
- Carter
- ServiceStack
- API Endpoints

To learn more about minimal APIs

thank you

@ladynaggaga