



How To Observe Your Blazor WebAssembly Application With OpenTelemetry And Real User Monitoring?

Harry Kimpel
@harrykimpel

whoami



Harry Kimpel

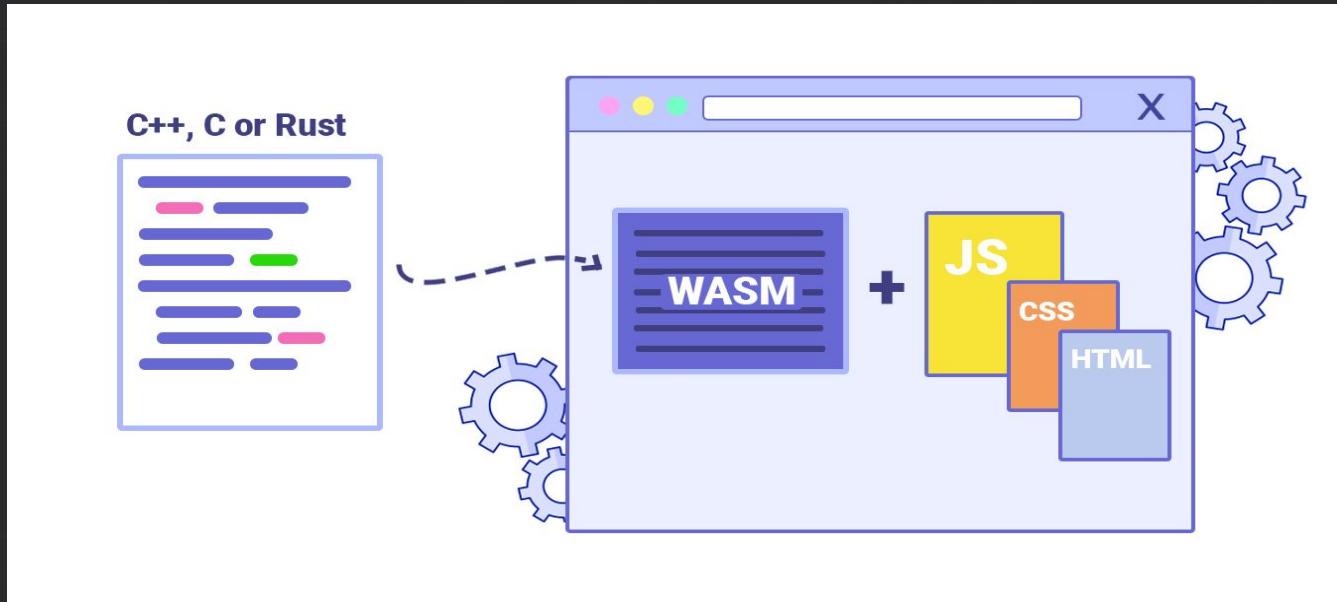
Principal Developer Relations Engineer

New Relic

@harrykimpel



Web Assembly



BYTECODE
ALLIANCE



WasmEdgeRuntime



.NET



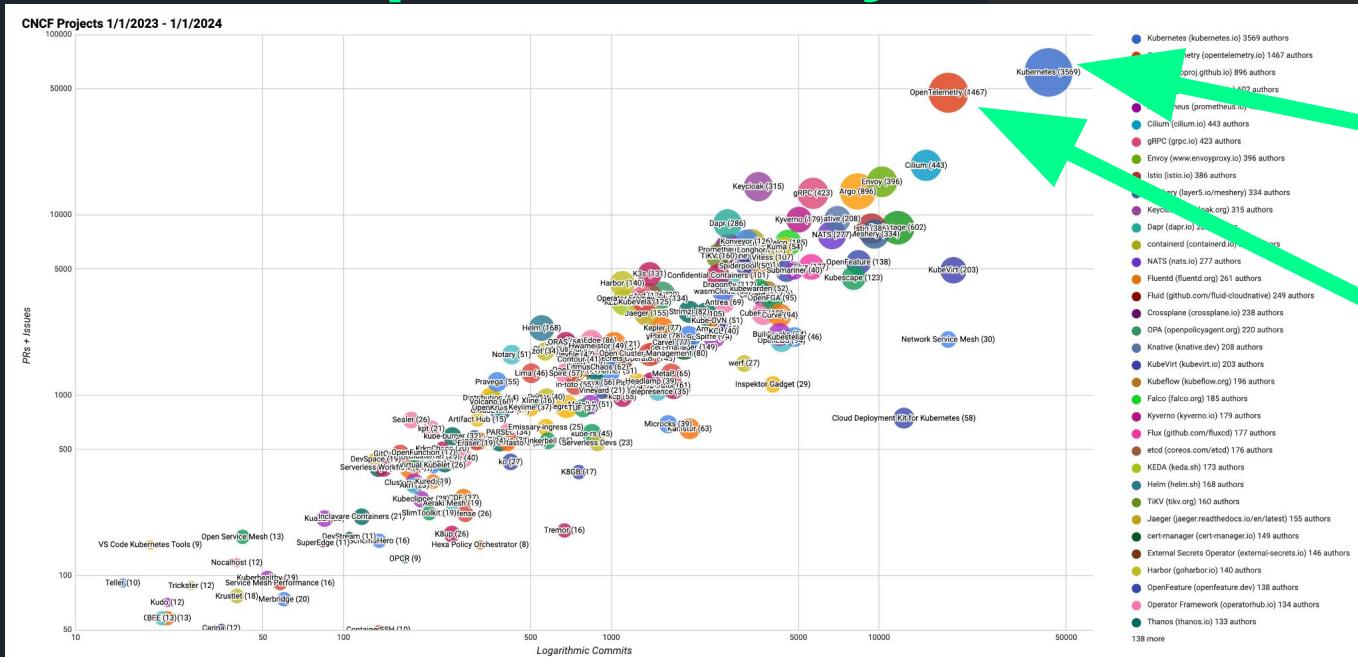
WA

Key Facts on OpenTelemetry

- OpenTelemetry is an **Incubating** project of CNCF.
- Formed through a merger of the **OpenTracing** and **OpenCensus** projects.
- **Vendor agnostic** - set of APIs, libraries, integrations, and a collector service for telemetry.
- **Standardizes** how you collect telemetry data from your applications and services.
- Send it to an **Observability platform** of your choice.



Key Facts on OpenTelemetry (CNCF Projects 1/1/2023 - 1/1/2024)



OpenTelemetry is second in CNCF - strong interest in modern **Observability**.

Enhanced support for **Open Standards**, including OpenTelemetry, eBPF, and Grafana - Gartner MQ 2022

.NET



WA





Open Telemetry with Blazor backend Service

OpenTelemetry with .NET backend

Current Status

[Docs](#) / [Language APIs & SDKs](#) / [.NET](#)

.NET

 A language-specific implementation of OpenTelemetry in .NET.

This is the OpenTelemetry .NET documentation. OpenTelemetry is an observability framework – an API, SDK, and tools that are designed to aid in the generation and collection of application telemetry data such as metrics, logs, and traces. This documentation is designed to help you understand how to get started using OpenTelemetry .NET.

Status and Releases

The current status of the major functional components for OpenTelemetry .NET is as follows:

[Traces](#) [Metrics](#) [Logs](#)

[Stable](#) [Stable](#) [Stable](#)

For releases, including the [latest release](#), see [Releases](#).

Opentelemetry with .NET backend

OpenTelemetry .NET Automatic Instrumentation



```
<Project Sdk="Microsoft.NET.Sdk.Web">

<ItemGroup>
  <PackageReference Include="Microsoft.AspNetCore.Identity.EntityFrameworkCore" Version="8.0.0" />
  <PackageReference Include="Microsoft.EntityFrameworkCore.InMemory" Version="8.0.0" />
  <PackageReference Include="NSwag.AspNetCore" Version="14.0.3" />
  <PackageReference Include="OpenTelemetry.Exporter.OpenTelemetryProtocol" Version="1.8.0" />
  <PackageReference Include="OpenTelemetry.Instrumentation.EntityFrameworkCore" Version="1.0.0-beta.9" />
  <PackageReference Include="OpenTelemetry.Instrumentation.Runtime" Version="1.8.0" />
  <PackageReference Include="OpenTelemetry" Version="1.8.0" />
  <PackageReference Include="OpenTelemetry.AutoInstrumentation.Runtime.Native" Version="1.5.0" />
  <PackageReference Include="OpenTelemetry.Exporter.Console" Version="1.8.0" />
  <PackageReference Include="OpenTelemetry.Extensions.Hosting" Version="1.8.0" />
  <PackageReference Include="OpenTelemetry.Instrumentation.AspNetCore" Version="1.8.1" />
  <PackageReference Include="OpenTelemetry.Instrumentation.Http" Version="1.8.1" />
</ItemGroup>

</Project>
```

The screenshot shows a Visual Studio Code interface with the following details:

- Title Bar:** BlazorWebAssemblyStandaloneWithIdentity
- Explorer:** Shows the project structure under "BLAZORWEBASSEMBLYST...":
 - > Backend
 - > BlazorWasmAuth
 - BlazorWasmAuth.sln
 - BlazorWasmAuth.slnLaunch.user
 - ① README.md
- Terminal:** Displays application logs:

```
info: Microsoft.EntityFrameworkCore.Update[30100]
      Saved 1 entities to in-memory store.
info: Microsoft.EntityFrameworkCore.Update[30100]
      Saved 3 entities to in-memory store.
info: Microsoft.EntityFrameworkCore.Update[30100]
      Saved 1 entities to in-memory store.
info: Microsoft.EntityFrameworkCore.Update[30100]
      Saved 2 entities to in-memory store.
info: Microsoft.Hosting.Lifetime[14]
      Now listening on: https://localhost:7211
info: Microsoft.Hosting.Lifetime[14]
      Now listening on: http://localhost:5266
info: Microsoft.Hosting.Lifetime[0]
      Application started. Press Ctrl+C to shut down.
info: Microsoft.Hosting.Lifetime[0]
      Hosting environment: Development
info: Microsoft.Hosting.Lifetime[0]
```
- Activity Trace:** A specific log entry is highlighted with a green border:

```
lonewithIdentity/Backend
Activity.TraceId: 5e199324748dc33fefef6be9ef87051551
Activity.SpanId: 6d09758bacac66ec
Activity.TraceFlags: Recorded
Activity.ActivitySourceName: Microsoft.AspNetCore
Activity.DisplayName: OPTIONS
Activity.Kind: Server
Activity.StartTime: 2024-05-06T16:24:05.7253440Z
Activity.Duration: 00:00:00.1363160
Activity.Tags:
  server.address: localhost
  server.port: 7211
  http.request.method: OPTIONS
  url.scheme: https
  url.path: /manage/info
  network.protocol.version: 2
  user_agent.original: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/124.0.0.0 Safari/537.36
  http.response.status_code: 204
Resource associated with Activity:
  service.name: BlazorWasmBackend
  service.instance.id: 03859d9a-2433-4c04-8bf7-128c93c79744
  deployment.environment: staging
  service.version: 1.0.0
  telemetry.sdk.name: opentelemetry
  telemetry.sdk.language: dotnet
  telemetry.sdk.version: 1.8.0
```
- Bottom Status Bar:** Shows icons for CodeStream, Live Share, Downloading .NET..., and a "Formatting" dropdown.

Backend > \$ run.sh

```
1 export NEW_RELIC_LICENSE_KEY=MY_NEW_RELIC_LICENSE_KEY
2 export OTEL_EXPORTER_OTLP_HEADERS=api-key=${NEW_RELIC_LICENSE_KEY}
3 export OTEL_EXPORTER_OTLP_ENDPOINT=https://otlp.nr-data.net
4 export OTEL_EXPORTER_OTLP_PROTOCOL=http/protobuf
5 export OTEL_SERVICE_NAME=BlazorWASMBackend
6 export OTEL_RESOURCE_ATTRIBUTES=deployment.environment=staging,service.version=1.0.0
7 export DOTNET_RUNTIME_ID="osx-x64"
8 export ENABLE_PROFILING=true
9 export OTEL_DOTNET_AUTO_HOME=/Users/hkimpel/.otel-dotnet-auto
10 export CORECLR_ENABLE_PROFILING=1
11 export CORECLR_PROFILER="{918728DD-259F-4A6A-AC2B-B85E1B658318}"
12 export DOTNET_RUNTIME_ID="osx-x64"
13 export SUFIX="dylib"
14 export CORECLR_PROFILER_PATH="$OTEL_DOTNET_AUTO_HOME/$DOTNET_RUNTIME_ID/OpenTelemetry.AutoInstrumentation.Native.$SUFIX"
15
16 dotnet run -lp https://bin/Release/net8.0/publish/Backend
17
```

品 Good ○ ⊕ Good< ⌚ from May 6, 06:34 pm to May 6, 06:55 pm (GMT+2) >[Summary](#)

MONITOR

[Distributed tracing](#)[Service map](#)[Transactions](#)[Databases](#)[External services](#)[Logs](#)

TRIAGE

[Errors \(errors inbox\)](#)[Diagnose](#)

EVENTS

[Change tracking](#)

DATA

[Metrics explorer](#)

SETTINGS

[Alert conditions](#)

MORE VIEWS

[Add app](#)[Events explorer](#)[Help](#)[Issues & activity](#)View charts by [Metrics](#) [Spans](#)Narrow data to Enter any attribute or value

Group by none

Compare with none

Response time

Since May 6, 06:34 pm until May 6, 06:55 pm

800 ms

700 ms

600 ms

500 ms

400 ms

300 ms

200 ms

100 ms

0 ms

6:35pm

6:40pm

6:45pm

6:50pm

...

● Response time

Throughput

Since May 6, 06:34 pm until May 6, 06:55 pm

50

45

40

35

30

25

20

15

10

5

0

...

Error rate

Since May 6, 06:34 pm until May 6, 06:55 pm

100%

90%

80%

70%

60%

50%

40%

30%

20%

10%

0%

...

Activity

Filters



No activity in this time range

To see more activity, try adjusting your filters or time range.

Related entities



You have gaps in your alert coverage

Some parts of your stack don't have alerts watching them.

[Set up alerts](#)

User Experience

2

Services

[See full map](#)[Add to dashboard](#)

BlazorWASMBackend [Tags](#) [Metadata](#) ... Good  Good

from May 6, 06:34 pm to May 6, 06:55 pm (GMT+2) >

 Summary

MONITOR

 Distributed tracing Service map Transactions Databases External services Logs

TRIAGE

 Errors (errors inbox) Diagnose

EVENTS

 Change tracking

DATA

 Metrics explorer

SETTINGS

 Alert conditions

MORE VIEWS

 Add app Events explorer Help Issues & activity

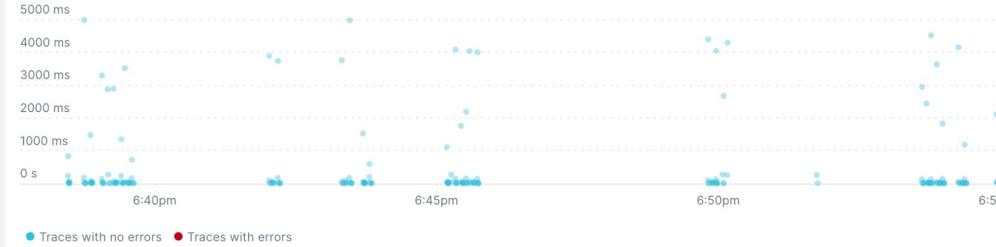
Search for traces by span attributes... (e.g. name = spanName)

 Refine

Trace groups Root entry span Root entity Errors

 Multi-span only

View by Root span duration



Plot	Trace name	Start time	Root span duration	Entities	Spans	Errors ↓
	OPTIONS BlazorWASMBackend	14 min ago May 6, 2024 18:...	118 µs	1	1	0
	OPTIONS BlazorWASMBackend	14 min ago May 6, 2024 18:...	109 µs	1	1	0
	OPTIONS BlazorWASMBackend	14 min ago May 6, 2024 18:...	87 µs	1	1	0
	GET localhost:7211/roles BlazorWASMFrontend	14 min ago May 6, 2024 18:...	4621 ms	2	2	0
	GET localhost:7211/manage/info BlazorWASMFrontend	14 min ago May 6, 2024 18:...	7 ms	2	2	0
	POST localhost:7211/login BlazorWASMFrontend	14 min ago May 6, 2024 18:...	115 ms	2	2	0
	OPTIONS BlazorWASMBackend	14 min ago May 6, 2024 18:...	79 µs	1	1	0

GET localhost:7211/roles

BlazorWebAssemblyStandaloneWithIdentityFrontend | May 6 at 6:45pm | Trace ID: [fdbc8270ba4fba513f605be43d1f21b8](#) | Spans: 2 [i](#)

Trace details Logs (0)

0 anomalous spans [▼](#)

0 errors [▼](#)

3 entities [▼](#)

Find spans by name or ID

Trace duration
4579.99 ms

Back-end duration
4019.99 ms

1

GET localhost:7211/roles

BlazorWASMFrontend

4.62 s

GET /roles

BlazorWASMBackend



4.61 s



BlazorWebAssemb...Frontend

1 CALLS



BlazorWASMBackend

1 CALLS



localhost



Expand all

Collapse all

Standard Manual



Focus on: Slow spans

Reset

Maximize

1

GET localhost:7211/roles

BlazorWebAssemblyStandaloneWithIdentityFrontend

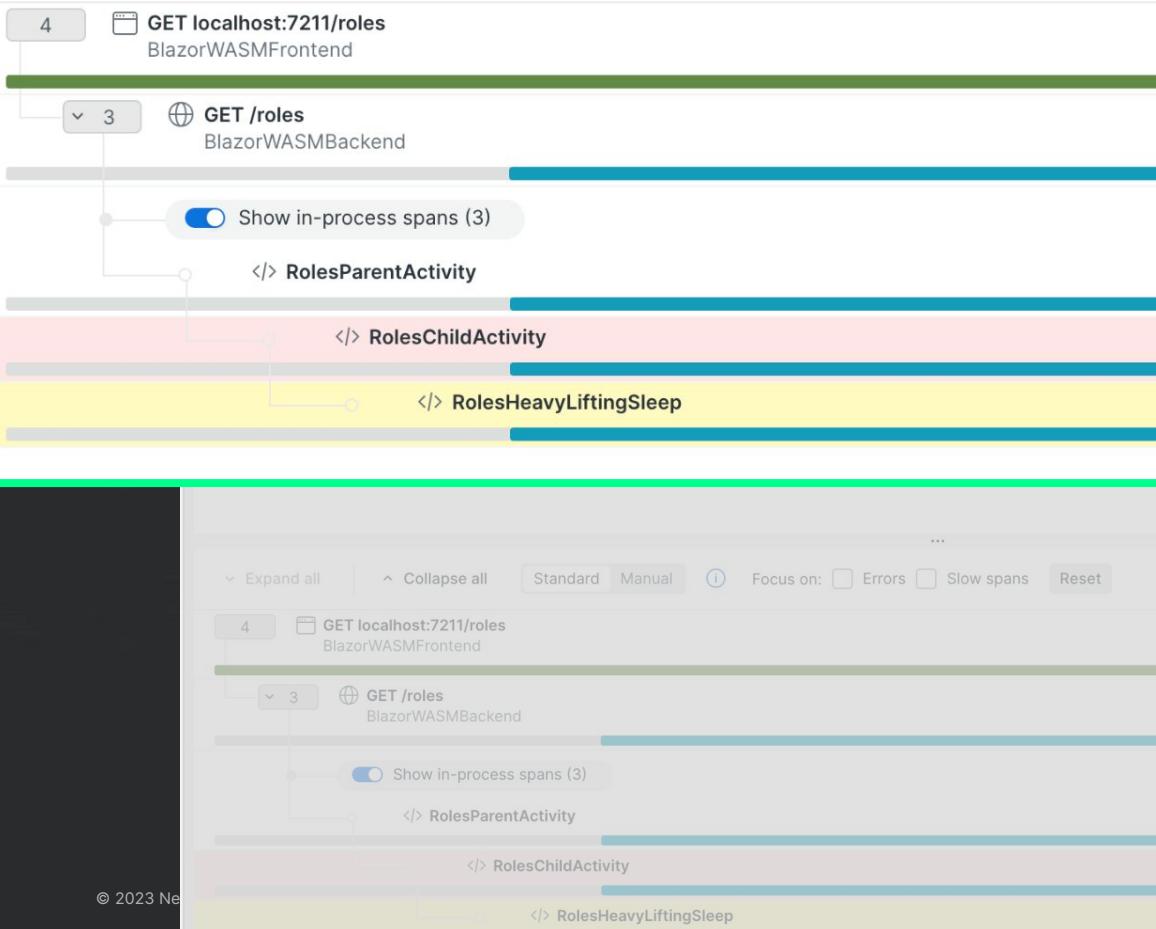
4.02 s

GET /roles

BlazorWASMBackend



4.02 s



RolesHeavyLiftingSleep

BlazorWASMBackend

Performance Attributes Details

2391.64 ms

DURATION

BlazorWASMBackend

service.version

1.0.0

simulatedWaitMsg
ChildActivity simulated wait (2390ms)simulatedWaitTimeMs
2390span.kind
internaltelemetry.sdk.language
dotnettelemetry.sdk.name
opentelemetrytelemetry.sdk.version
1.8.0timestamp
1715017304702

.NET



Open Telemetry with Blazor WASM App

Opentelemetry with .NET Blazor WASM

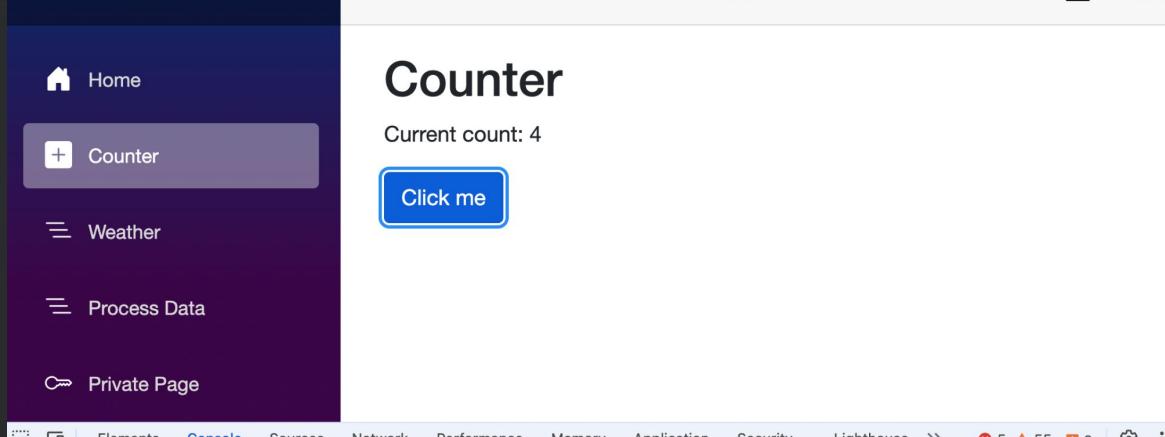
OpenTelemetry .NET Automatic Instrumentation for Blazor



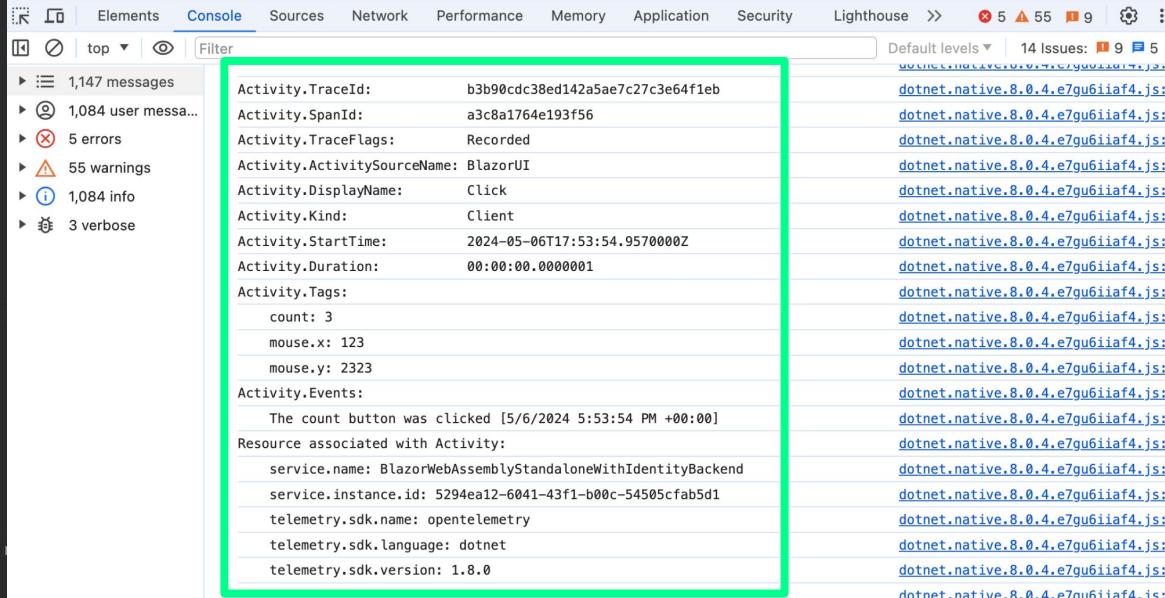
```
<Project Sdk="Microsoft.NET.Sdk.BlazorWebAssembly">

  <ItemGroup>
    <PackageReference Include="Microsoft.AspNetCore.Components.WebAssembly.Authentication" Version="8.0.0" />
    <PackageReference Include="Microsoft.Extensions.Http" Version="8.0.0" />
    <PackageReference Include="Microsoft.AspNetCore.Components.WebAssembly" Version="8.0.0" />
    <PackageReference Include="Microsoft.AspNetCore.Components.WebAssembly.DevServer" Version="8.0.0" />
    <PrivateAssets="all" />
    <PackageReference Include="OpenTelemetry.Exporter.OpenTelemetryProtocol" Version="1.8.0" />
    <PackageReference Include="OpenTelemetry.Instrumentation.Runtime" Version="1.8.0" />
    <PackageReference Include="OpenTelemetry" Version="1.8.0" />
    <PackageReference Include="OpenTelemetry.AutoInstrumentation.Runtime.Native" Version="1.5.0" />
    <PackageReference Include="OpenTelemetry.Exporter.Console" Version="1.8.0" />
    <PackageReference Include="OpenTelemetry.Extensions.Hosting" Version="1.8.0" />
  </ItemGroup>

</Project>
```



The screenshot shows a Blazor application interface. On the left is a sidebar with navigation links: Home, Counter (which is highlighted with a blue background), Weather, Process Data, and Private Page. The main content area has a title "Counter" and displays the message "Current count: 4" above a blue button labeled "Click me".



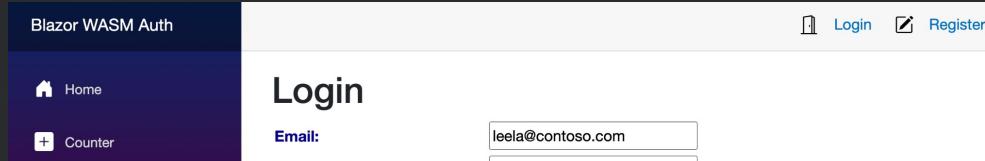
The screenshot shows the New Relic browser extension's Console tab. The left sidebar lists metrics: 1,147 messages, 1,084 user messages, 5 errors, 55 warnings, 1,084 info, and 3 verbose. The main pane displays detailed activity logs for a click event. A green box highlights the activity details for the click:

```
Activity.TraceId: b3b90cdc38ed142a5ae7c27c3e64f1eb
Activity.SpanId: a3c8a1764e193f56
Activity.TraceFlags: Recorded
Activity.ActivitySourceName: BlazorUI
Activity.DisplayName: Click
Activity.Kind: Client
Activity.StartTime: 2024-05-06T17:53:54.957000Z
Activity.Duration: 00:00:00.000001
Activity.Tags:
  count: 3
  mouse.x: 123
  mouse.y: 2323
Activity.Events:
  The count button was clicked [5/6/2024 5:53:54 PM +00:00]
Resource associated with Activity:
  service.name: BlazorWebAssemblyStandaloneWithIdentityBackend
  service.instance.id: 5294ea12-6041-43f1-b00c-54505cfab5d1
  telemetry.sdk.name: opentelemetry
  telemetry.sdk.language: dotnet
  telemetry.sdk.version: 1.8.0
```

The right side of the console shows a list of native .NET library calls, all of which are identical: `dotnet.native.8.0.4.e7gu6liaf4.js:8`.

Opentelemetry with .NET Blazor WASM

Limitations with Blazor WebApp



✖ ► crit: Microsoft.AspNetCore.Components.WebAssembly.Rendering.WebAssemblyRenderer[100]
Unhandled exception rendering component: Operation is not supported on this platform.
System.PlatformNotSupportedException: Operation is not supported on this platform.

```
    > c: Microsoft.AspNetCore.Components.WebAssembly.Rendering.WebAssemblyRenderer[100]
      Unhandled exception rendering component.
      Operation is not supported on this platform.
      System.PlatformNotSupportedException: Operation is not supported on this platform
        at System.Threading.ThreadStart.<Run>B7N<ThreadStart>(Boolean internalhread)
        at System.Threading.Thread.Start(Boolean cancellationToken, Boolean internalThread)
        at System.Threading.Thread.Start()
      at OpenTelemetry.Batch.ExporterProcessor[1](System.Diagnostics.Activity, System.Diagnostics.DiagnosticSource, Version=0.0.0, Culture=neutral, PublicKeyToken=cc1bf3cd2dd501c1).ctor(BaseExporter`1 exporter, Int32 maxQueueSize, Int32 scheduledDelayMilliseconds, Int32 exportTimeOutMilliseconds)
      at OpenTelemetry.Batch.ActiveExporterProcessor..ctor(BaseExporter`1 exporter, Int32 maxQueueSize, Int32 scheduledDelayMilliseconds, Int32 exportTimeOutMilliseconds, Int32 maxExportBatchSize)
      at OpenTelemetry.Trace.OtpTraceExporterHelperExtensions.BuildOtpTraceExporterProcessor(IServiceProvider serviceProvider, OtpExporterOptions exporterOptions, SdkLimitOptions sdkLimitOptions, ExperimentalOptions experimentalOptions, ExporterProcessorType exporterProcessorType, BatchActiveExporterProcessorOptions batchActiveExporterProcessorOptions, Func<IExporter, IExporter> exporterBuilder)
      at OpenTelemetry.Trace.OtpTraceExporterHelperExtensions.BuildOtpTraceExporterProcessor(IServiceProvider serviceProvider, OtpExporterOptions exporterOptions, SdkLimitOptions sdkLimitOptions, ExperimentalOptions experimentalOptions, Func<T> configureExporterInstance)
      at OpenTelemetry.Trace.OtpTraceExporterHelperExtensions.<__DisplayClass2_0.>AddOtpExporter[b_0](IServiceProvider sp)
      at OpenTelemetry.Trace.TracerProviderBuilderExtensions.<>c__DisplayClass8_0.>AddProcessor[b_0](IServiceProvider sp, TracerProviderBuilder builder)
      at OpenTelemetry.Trace.OpenTelemetryDependencyInjectionTracingServiceCollectionExtensions.ConfigureTracerProviderBuilderCallbackWrapper.ConfigureBuilderv(IServiceProvider serviceProvider, TracerProviderBuilder tracerProviderBuilder)
      at OpenTelemetry.Trace.TracerProviderSdk..ctor(IServiceProvider serviceProvider, Boolean ownsServiceProvider)
      at OpenTelemetry.Trace.TracerProviderBuilderBase..ctor(IServiceProvider serviceProvider)
      at OpenTelemetry.Trace.TracerProviderBuilderBase..ctor(IServiceProvider serviceProvider, ILoggerBuilder)
      at OpenTelemetry.Trace.TracerProviderBuilderExtensions.Build(TracerProviderBuilder tracerProviderBuilder)
      at BlazorWasAuth.Identity.CookieAuthenticationServiceProvider.LoginAsRazor(string email, string password, ILogger logger) in /Users/khimpel/projects/dotnet8/dotnet8/dotnet-blazor-samples/8/BlazorWebAssemblySamples/BlazorWasAuth/Identity/CookieAuthenticationServiceProvider.cs:line 140
      at BlazorWasAuth.Components.Identity.Login.RazorIdentityIn /Users/khimpel/projects/dotnet8/dotnet-blazor-samples/8/BlazorWebAssemblySamples/BlazorWasAuth/Components/Identity/Login/razor/line 74
      at Microsoft.AspNetCore.Components.ComponentBase.CallStateHasChangedOnAsyncCompletion(Task task)
      at Microsoft.AspNetCore.Components.RenderTree.Renderer.GetErrorHandledTask(Task taskToHandle, ComponentState owningComponentState)
```

Opentelemetry with .NET Blazor WASM

Real User Monitoring is coming ...

[Docs](#) / [Language APIs & SDKs](#) / [JavaScript](#) / [Getting Started](#) / Browser

Browser

Learn how to add OpenTelemetry to your browser app

Warning

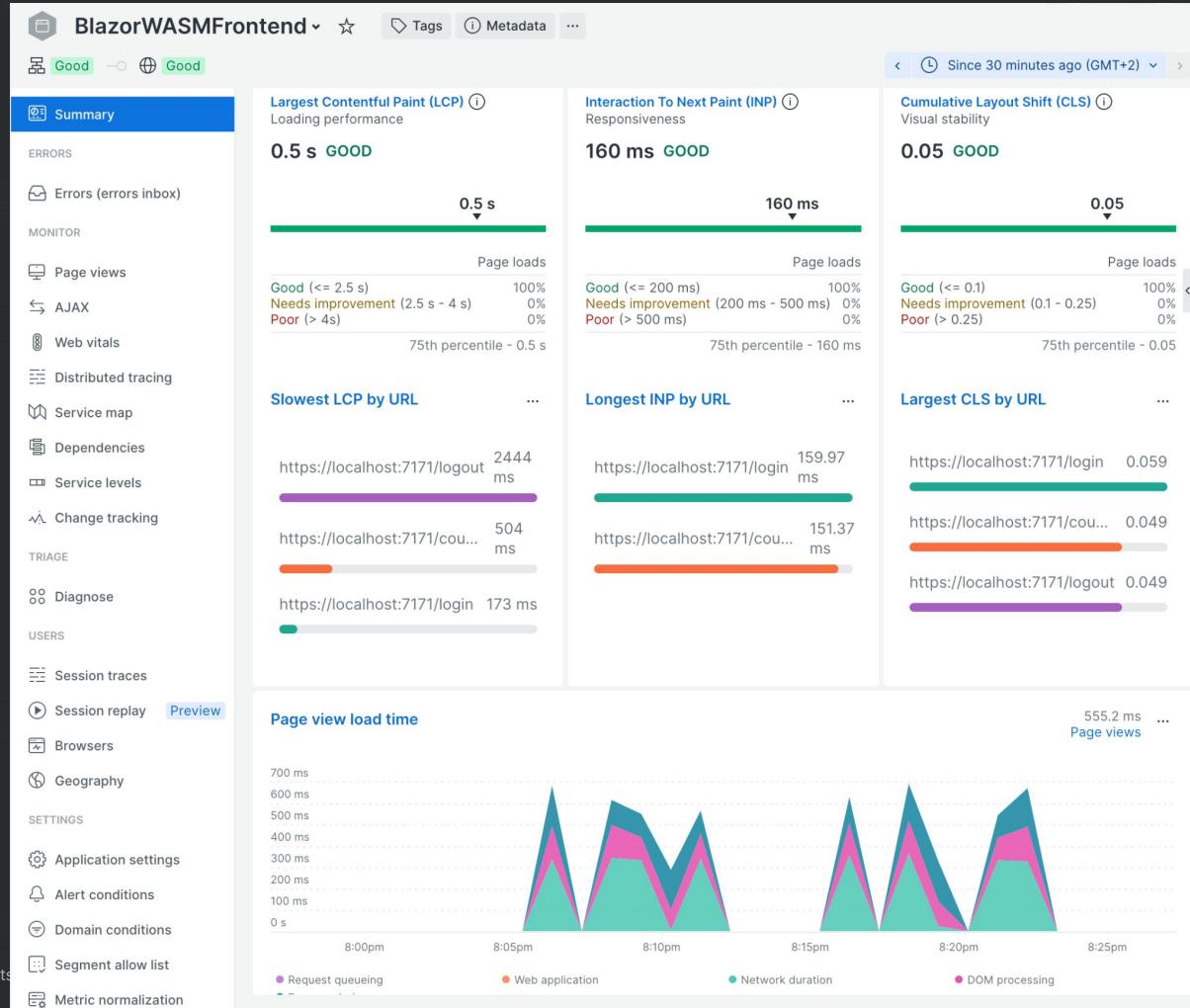
Client instrumentation for the browser is **experimental** and mostly **unspecified**. If you are interested in helping out, get in touch with the [Client Instrumentation SIG](#).

While this guide uses the example application presented below, the steps to instrument your own application should be similar.

Prerequisites

Ensure that you have the following installed locally:

- [Node.js](#)
- [TypeScript](#), if you will be using TypeScript.



BlazorWASMFrontend Good Tags Metadata ...

Since 30 minutes ago (GMT+2)

Summary

ERRORS

Errors (errors inbox)

MONITOR

Page views

AJAX

Web vitals

Distributed tracing

Service map

Dependencies

Service levels

Change tracking

TRIAGE

Diagnose

USERS

Session traces

Session replay Preview

Browsers

Geography

SETTINGS

Application settings

Alert conditions

Domain conditions

Segment allow list

Search for traces by span attributes... (e.g. name = spanName)

Trace groups Root entry span Root entity Errors

Multi-span only

View by Root span duration

5000 ms
4000 ms
3000 ms
2000 ms
1000 ms
0 s

8:10pm 8:15pm 8:20pm

Traces with no errors Traces with errors

Plot Trace name Start time ↓ Root span d... Enti... Spa... Errors

●	GET localhost:7211/manage/info BlazorWASMFrontend	7 min ago May 6, 2024 20:...	236 ms	2	2	0
●	GET localhost:7171/_framework/* BlazorWASMFrontend	7 min ago May 6, 2024 20:...	77 ms	1	1	0
●	GET localhost:7171/_framework/* BlazorWASMFrontend	7 min ago May 6, 2024 20:...	75 ms	1	1	0
●	GET localhost:7171/appsettings.json BlazorWASMFrontend	7 min ago May 6, 2024 20:...	27 ms	1	2	0
●	GET localhost:7171/_framework/* BlazorWASMFrontend	7 min ago May 6, 2024 20:...	74 ms	1	2	0
●	GET localhost:7211/manage/info BlazorWASMFrontend	9 min ago May 6, 2024 20:...	236 ms	2	2	0
●	GET localhost:7171/_framework/* BlazorWASMFrontend	9 min ago May 6, 2024 20:...	114 ms	1	1	0

© 2023 New Relic, Inc. All rights reserved.

 new relic



Good

Good

Summary

ERRORS

Errors (errors inbox)

MONITOR

Page views

AJAX

Web vitals

Distributed tracing

Service map

Dependencies

Service levels

Change tracking

TRIAGE

Diagnose

USERS

Session traces

Session replay Preview

Browsers

Geography

SETTINGS

Application settings

Alert conditions

Domain conditions

Segment allow list

Metric normalization

Desktop Mobile Tablet

Sort by Total time percentage

Group by requestUrl

Showing top 20

AJAX requests Total...

localhost/roles ... 72%

GET /roles

localhost/manage... 2.32%

GET /manage/info

localhost/_framew... 1.32%

GET /_framework/...

localhost/_framew... 1.18%

GET /_framework/...

localhost/_framew... 1.13%

GET /_framework/...

localhost/_framew... 0.82%

GET /_framework/...

localhost/_framew... 0.64%

GET /_framework/...

localhost/_framew... 0.61%

GET /_framework/...

localhost/_framew... 0.61%

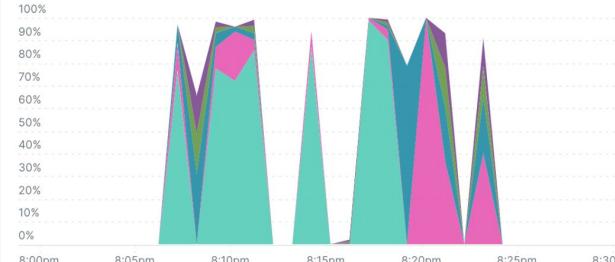
GET /_framework/...

localhost/_framew... 0.61%

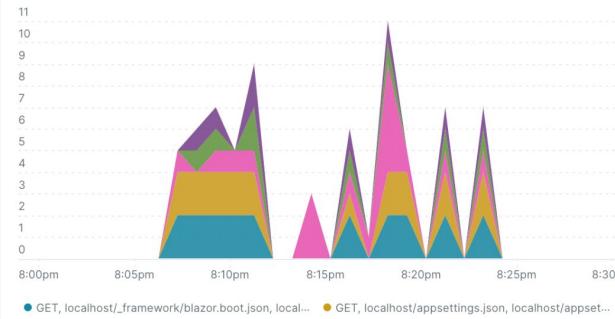
GET /_framework/...

View distributed traces

Total time percentage



Throughput (requests per minute)



 [Home](#) [Counter](#) [Weather](#) [Process Data](#)

Counter

Current count: 0

 [Click me](#)

```
1 <!DOCTYPE html>
2 <html lang="en">
3
4 <head>
5   <meta charset="utf-8" />
6   <meta name="viewport" content="width=device-width, initial-scale=1.0" />
7   <script src="newrelic.js"></script>
8   <title>Blazor WASM Auth</title>
9   <base href="/" />
10  <link rel="stylesheet" href="css/bootstrap/bootstrap.min.css" />
11  <link rel="stylesheet" href="css/app.css" />
12  <link rel="icon" type="image/png" href="favicon.png" />
13  <link href="BlazorWasmAuth.styles.css" rel="stylesheet" />
14 </head>
15
16 <body>
17   <div id="app">
18     <svg class="loading-progress">
19       <circle r="40%" cx="50%" cy="50%" />
20       <circle r="40%" cx="50%" cy="50%" />
21     </svg>
22     <div class="loading-progress-text"></div>
23   </div>
24
25   <div id="blazor-error-ui" data-nosnippet>
26     An unhandled error has occurred.
27     <a href="" class="reload">Reload</a>
28     <a class="dismiss">✖</a>
29   </div>
30   <script src="\_framework/blazor.webassembly.js"></script>
31 </body>
32
33 </html>
```

≡ Counter.razor ×

BlazorWasmAuth > Components > Pages > ≡ Counter.razor

```
1  @page "/counter"
2  @inject IJSRuntime JS
3
4  <script type="text/javascript">
5  function addNewRelicPageAction(currentCount) {
6    newrelic.addPageAction('counterClicked', {currentCount: currentCount});
7    return "success";
8  }
```

```
· private void IncrementCount()
· {
·     ···· currentCount++;
·     ···· Trace(currentCount);
·     ···· JS.InvokeAsync<string>("addNewRelicPageAction", currentCount);
· }
```

```
23
24  @code {
25      ···· private int currentCount = 0;
26
27      ···· private void IncrementCount()
28      {
29          ···· currentCount++;
30          ···· Trace(currentCount);
31          ···· JS.InvokeAsync<string>("addNewRelicPageAction", currentCount);
32      }
}
```

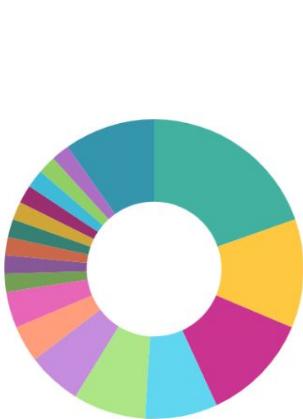
+ Add variable



Search for any attribute or value.

Click Counter distribution

May 7, 1:53am – May 7, 3:58am



counts (51)

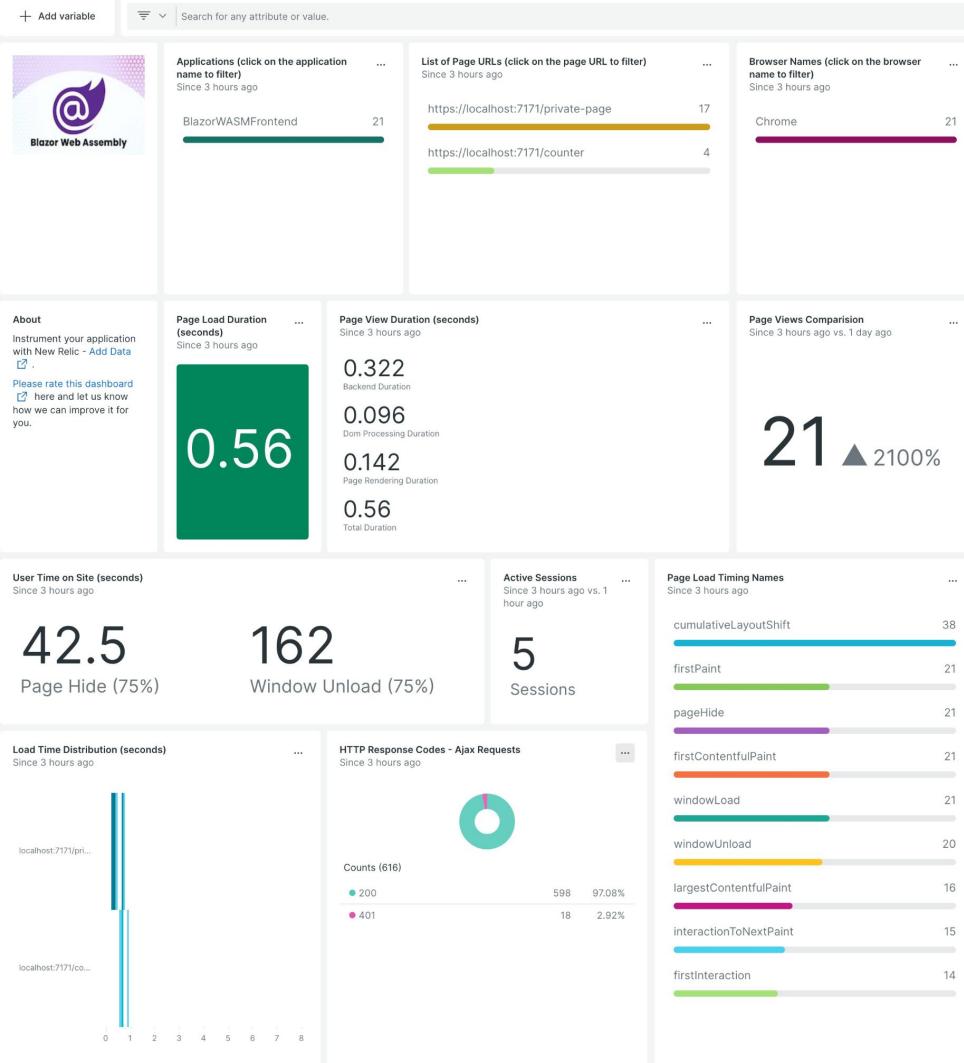
● 1	10	19.61%
● 2	6	11.76%
● 3	6	11.76%
● Other	5	9.8%
● 4	4	7.84%
● 5	4	7.84%
● 6	3	5.88%
● 7	2	3.92%
● 8	2	3.92%
● 10	1	1.96%
● 11	1	1.96%

Click Counter distribution

May 7, 1:54am – May 7, 3:58am



● 1 ● 2 ● 3 ● 4 ● 5 ● 6 ● 7 ● 8 ● 10 ● 11 ● 12 ● 13 ● 14 ● 15 ● 16 ● 17 ● 9



References

- [Build a full-stack web app with Blazor](#)
- OpenTelemetry
 - [.NET](#)
 - [Other languages](#)
 - [Real User Monitoring](#)
- New Relic
 - [Real User Monitoring](#)
 - [New Relic Browser APIs](#)
- Blazor WebAssembly repo

Slides available here:



A graphic element consisting of a vertical bar with a pink and purple striped pattern, resembling a fingerprint or a barcode.

THANK YOU

