

Product or feature is in a pre-release state and might change or have limited support. For more information, see the [product launch stages](#) (/products/#product-launch-stages).

You can enable Trace for ASP.NET core applications by using the `Google.Cloud.Diagnostics.AspNetCore` client library for C#.

Follow the instructions at [nuget.org](https://www.nuget.org) (<https://www.nuget.org/packages/Google.Cloud.Diagnostics.AspNetCore/2.0.1>) to install the library.

After the `Google.Cloud.Diagnostics.AspNetCore` client library is installed, complete the following steps to enable and configure Trace:

1. Add the following `using` directives to your application's `Startup.cs` file:

```
using Google.Cloud.Diagnostics.AspNetCore;
using Microsoft.AspNetCore.Builder;
using Microsoft.AspNetCore.Hosting;
using Microsoft.Extensions.Configuration;
using Microsoft.Extensions.DependencyInjection;
using Microsoft.Extensions.Logging;
using Microsoft.Extensions.Options;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Threading.Tasks;
```

2. Set the Google Cloud project ID that is passed to the `Google.Cloud.Diagnostics.AspNetCore` client library by entering your Google Cloud project ID in the `appsettings.json` file. In the following sample, replace `YOUR-GOOGLE-PROJECT-ID` with your Google Cloud project ID or number:

```
{
  "Logging": {
    "LogLevel": {
      "Default": "Information",
      "Microsoft": "Warning",
      "Microsoft.AspNetCore.Diagnostics": "Error"
    }
  },
  "AllowedHosts": "*",
  "GoogleCloudProjectId": "YOUR-GOOGLE-PROJECT-ID"
}
```

3. Create a new file named `StackdriverOptions.cs` with the following content:

```
/GoogleCloudPlatform/dotnet-docs-samples/blob/master/appengine/flexible/Stackdriver/StackDriverOptions.cs)
```

4. Create or edit the `ConfigureServices` method in your application's `Startup.cs` file so that it calls `services.AddGoogleTrace()`:

github.com/GoogleCloudPlatform/dotnet-docs-samples/blob/master/appengine/flexible/Stackdriver/Startup.cs

5. Edit the `Configure` method in your application's `Startup.cs` file and call `app.UseGoogleTrace()` *before* calling `app.UseStaticFile()` and `app.UseMvc()`.

github.com/GoogleCloudPlatform/dotnet-docs-samples/blob/master/appengine/flexible/Stackdriver/Startup.cs

Trace is now configured to sample the HTTP requests received by your application. To initiate traces from your application, see [sample application for ASP.NET core \(#aspnetcore\)](#).

You can use Trace on Google Cloud and when your application runs outside of Google Cloud.

When your application is running on Google Cloud, your application is automatically authenticated and you don't need to provide authentication credentials. However, you do need to ensure that your Google Cloud platform has the [Stackdriver Trace API access scope](#) (<https://developers.google.com/identity/protocols/googlescopes#cloudtracev2>) enabled.

For the following configurations, the default settings for the access scopes have the Stackdriver Trace API enabled:

- App Engine flexible environment
- Google Kubernetes Engine
- Compute Engine

Trace isn't supported for C# on App Engine standard environment.

If you use custom access scopes, then you must ensure that [Stackdriver Trace API access scope](#) (<https://developers.google.com/identity/protocols/googlescopes#cloudtracev2>) enabled. For `gcloud` users, specify access scopes using the `--scopes` flag and include the `trace.append` Stackdriver Trace API

access scope. For example, to create a GKE cluster with only the Stackdriver Trace API enabled, do the following:

When your application is running outside of Google Cloud, you must provide authentication credentials in the form of a service account to the client library. The service account must contain the [Cloud Trace agent role](/trace/docs/iam#roles) (/trace/docs/iam#roles). For instructions, see [Creating a service account](/iam/docs/creating-managing-service-accounts) (/iam/docs/creating-managing-service-accounts).

Google Cloud client libraries use [Application default credentials \(ADC\)](/docs/authentication/production) (/docs/authentication/production) to find your application's credentials. You provide these credentials by setting the `GOOGLE_APPLICATION_CREDENTIALS` environment variable:

```
CloudPlatform/dotnet-docs-samples/blob/master/appengine/flexible/Stackdriver/Controllers/TraceController.cs)
```

After deployment, you can view the traces in the Cloud Console Trace Viewer.

[Go to the Trace Viewer page \(https://console.cloud.google.com/traces/overview\)](https://console.cloud.google.com/traces/overview)

- **GitHub: Google.Cloud.Diagnostics.AspNetCore**
(<http://googleapis.github.io/google-cloud-dotnet/docs/Google.Cloud.Diagnostics.AspNetCore/index.html>)
- **NuGet: Google.Cloud.Diagnostics.AspNetCore package**
(<https://www.nuget.org/packages/Google.Cloud.Diagnostics.AspNetCore/>)
- **Source code** (<https://github.com/GoogleCloudPlatform/dotnet-docs-samples/tree/master/trace/api>)
- **GitHub issue tracker** (<https://github.com/googleapis/google-cloud-dotnet/issues>)
- **Stack Overflow** (<https://stackoverflow.com/questions/tagged/google-cloud-trace>)