

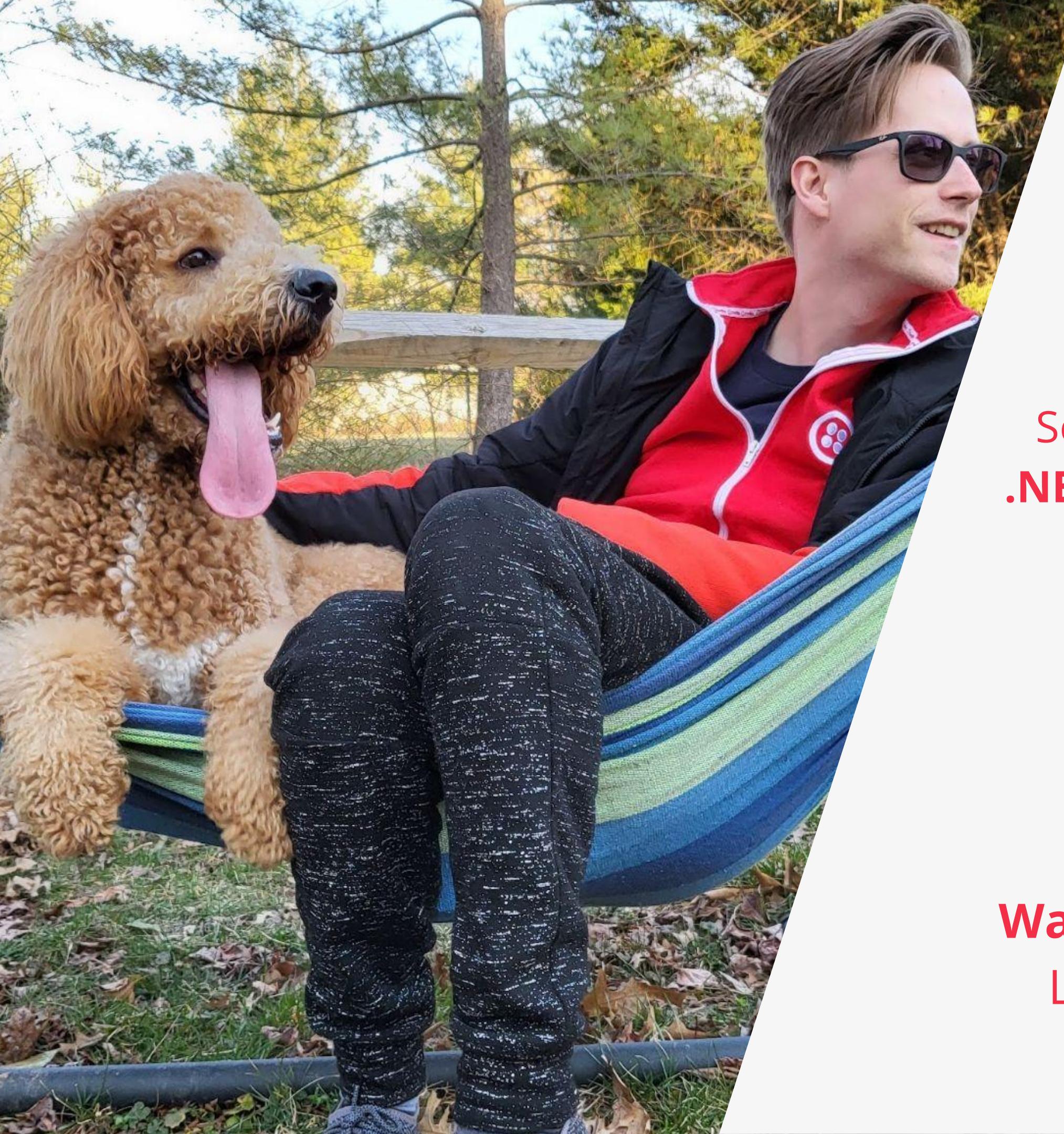


Develop webhooks locally with tunnels



Niels Swimberghe

Software Engineer, .NET Editor @ Twilio Developer Voices
Microsoft MVP



Niels Swimberghe

aka Swimbburger (He/Him)



Software Engineer, Technical Content
.NET Editor for Twilio Developer Voices



Twilio Blog: twilio.com/blog

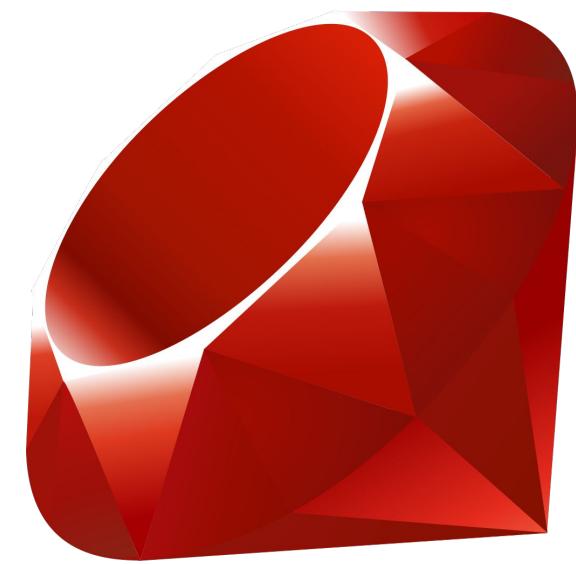
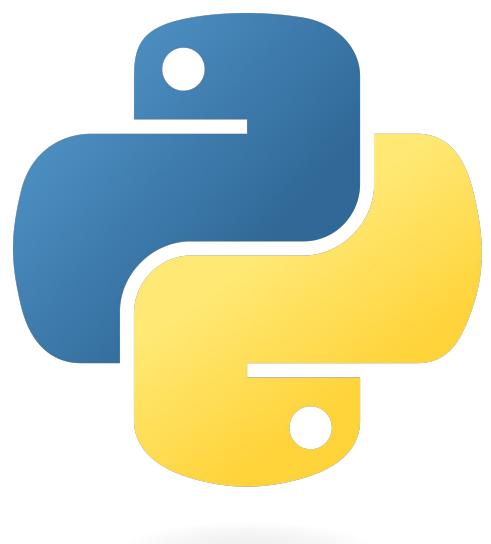
Personal blog: swimburger.net

Twitter: [@RealSwimburger](https://twitter.com/RealSwimburger)



Want to write for the Twilio blog?
Learn more at twilio.com/voices







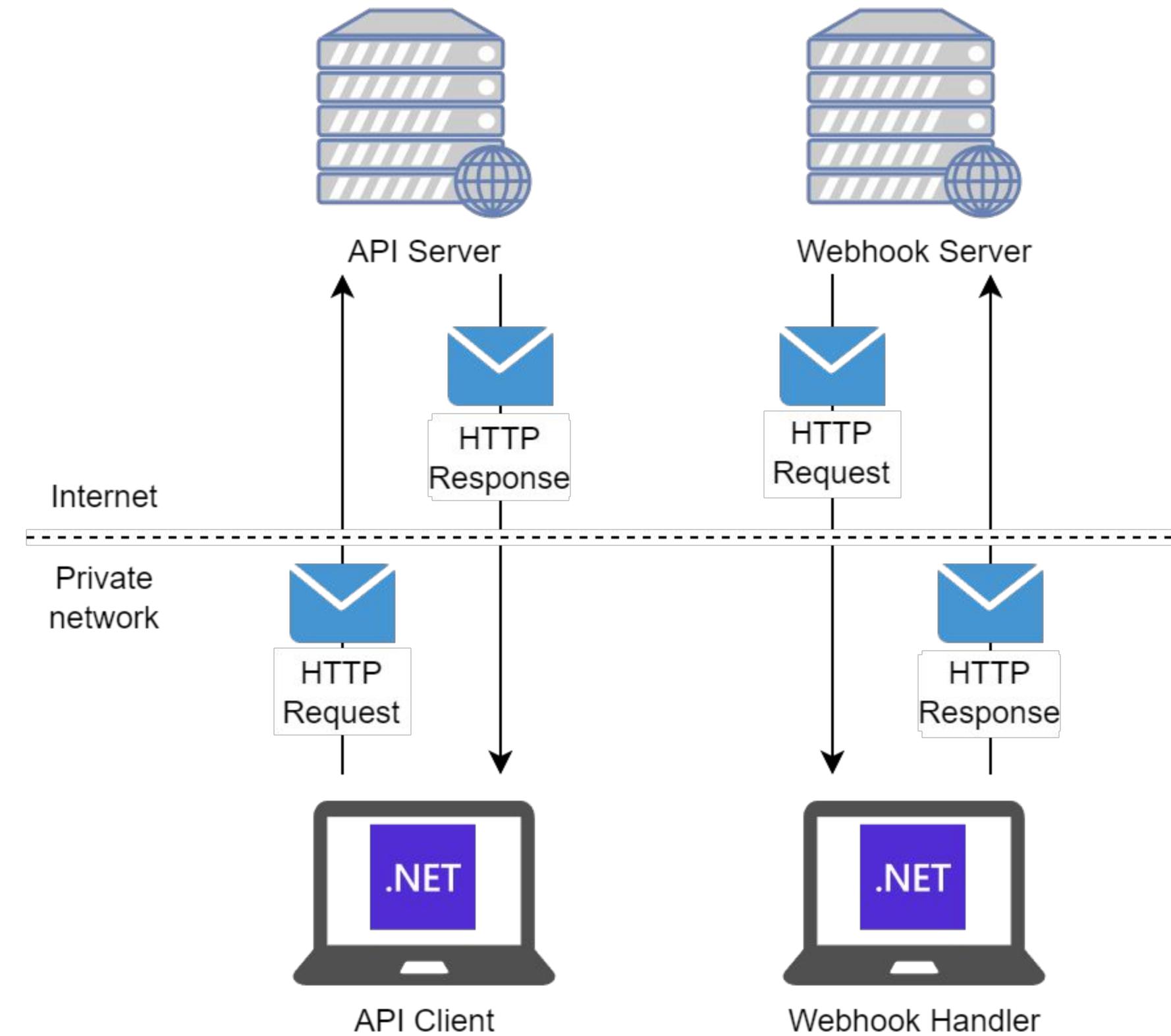


Agenda

- 01 What are webhooks?
- 02 Develop webhooks locally with Visual Studio dev tunnels
- 03 Automatically configure webhook
- 04 Securing your webhooks
- 05 Alternatives to VS dev tunnels



What are webhooks?





What are webhooks?

APIs (REST, gRPC, etc.) let you pull data from an external system by sending HTTP requests, and you receive the data in the HTTP response. You initiate the connection.

Webhooks let you, the user, define where an external system sends an HTTP request when certain events occur, containing the event data. The external system initiates the connection.



Webhooks are user-defined HTTP callbacks



Webhook example: GitHub

GitHub has webhooks for every type of event that can occur in your repository, account, or organization: push, pull requests, issues, workflows, etc.

Use cases:

- Automatically assign users to newly created issues or pull requests
- On new pull requests, ask contributors to sign agreement
- When a workflow fails, send a message to Teams or Slack



See all webhook: <https://bit.ly/GhHooks>

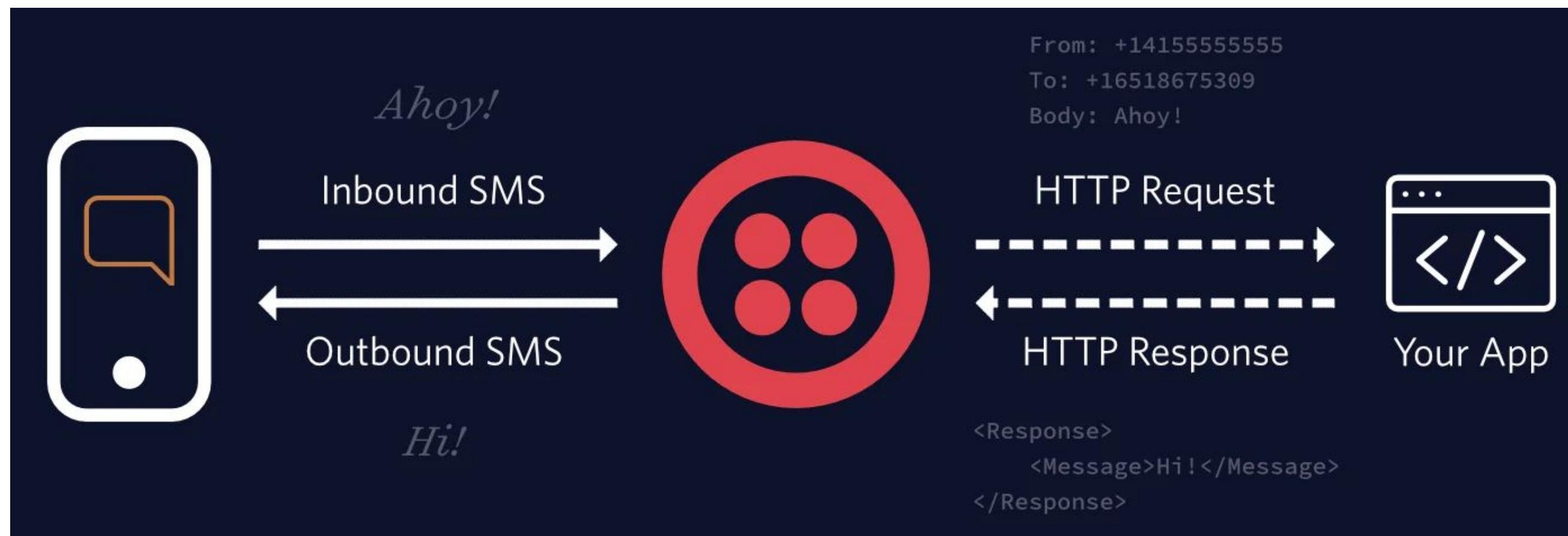


Twilio SMS & voice webhooks

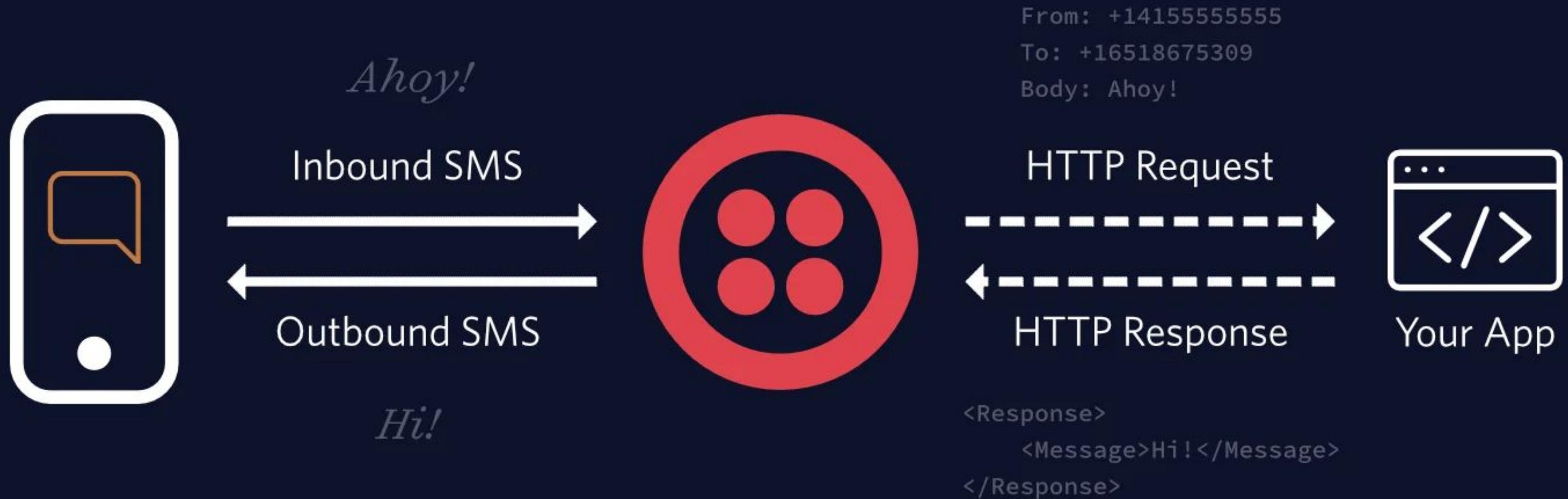
Twilio offers **programmable phone numbers**.

Using these phone number, you can **send SMS** and **make calls** using the Twilio **API**, but to **receive SMS** and **phone calls**, you need to use **webhooks**.

You configure the webhook URL and when the phone number receives a call or text, Twilio will send that information to the URL as an HTTP POST or GET request.



Demo: Twilio SMS webhooks





Twilio SMS & voice webhooks

Some webhooks expect you to respond in a specific way.

Twilio SMS and voice webhooks expect you to respond with TwiML instructions.
TwiML are a set of instructions in XML that Twilio executes for you.

TwiML Message - □ ×

```
<?xml version="1.0" encoding="utf-8"?>
<Response>
  <Message>Ahoy!</Message>
</Response>
```

TwiML Voice - □ ×

```
<?xml version="1.0" encoding="utf-8"?>
<Response>
  <Say>Ahoy!</Say>
  <Gather action="/ProcessGather" method="POST">
    <Say>How are you doing?</Say>
  </Gather>
</Response>
```

Learn more about TwiML: <https://bit.ly/WhatsTwiml>



Want to participate in the demos?

Your number and email address will be removed after the talk.

Text

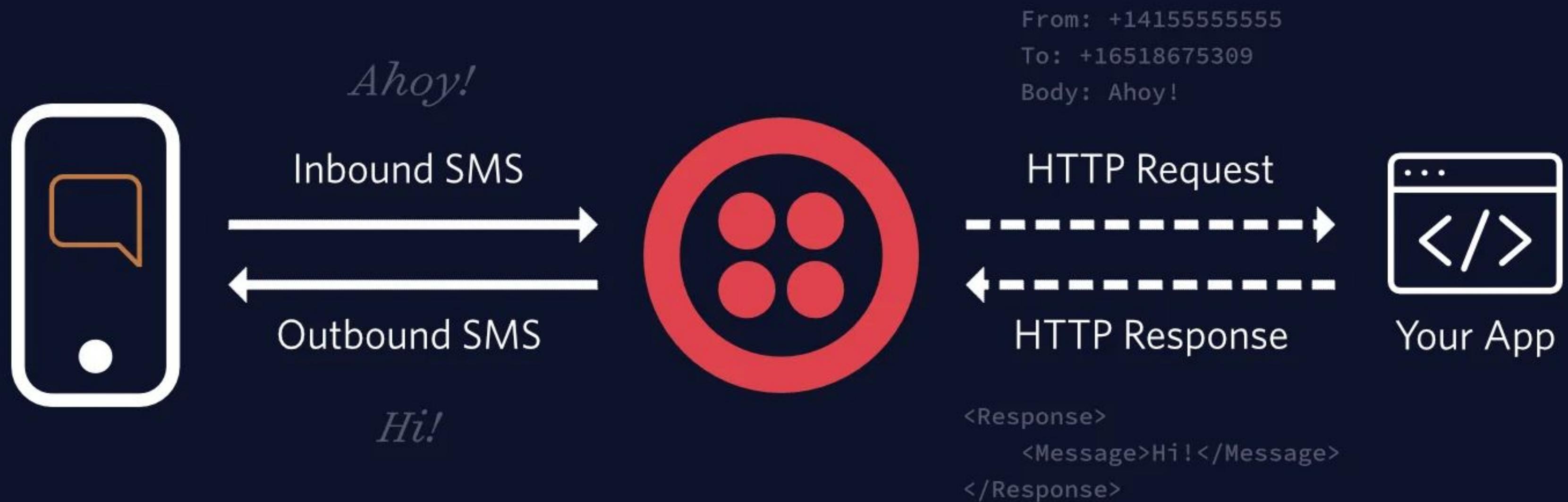


+1 (608) 889-3364

Call

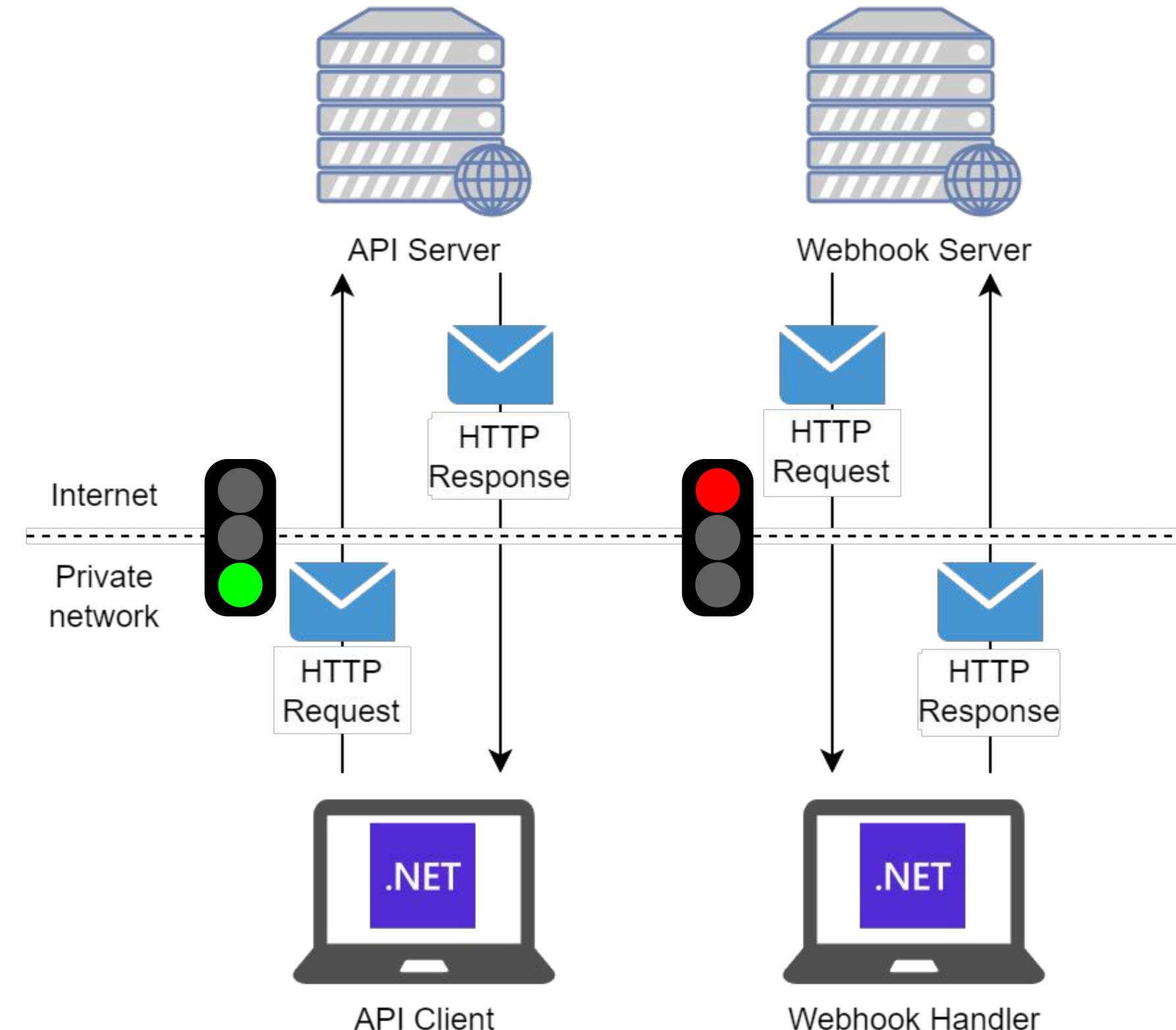


Demo: Twilio SMS webhooks





Problems with webhooks





Problems with webhooks

01. Webhooks need to be **publicly accessible** on the **internet**.
How do you develop webhooks on your local machine?

- 02.



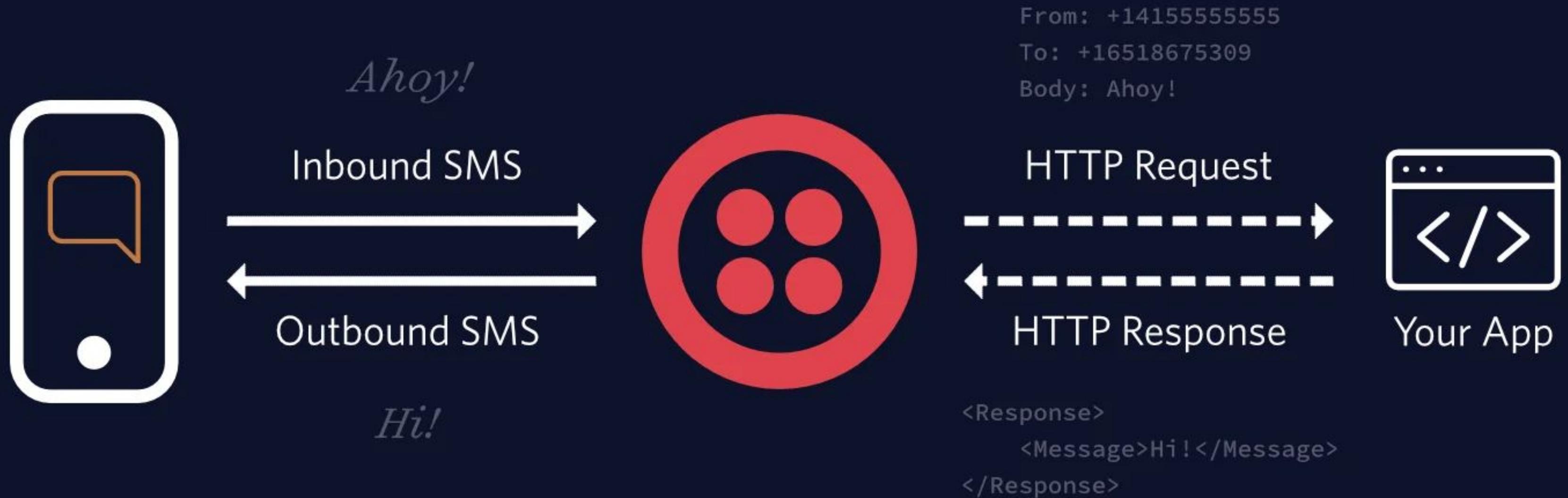
How to develop webhooks on localhost?

01. **Simulate** the webhook HTTP requests using scripts or the Twilio CLI webhook plugin

- 02.



Demo: webhook simulation





How to develop webhooks on localhost?

01. **Simulate** the webhook HTTP requests using scripts or the Twilio CLI webhook plugin



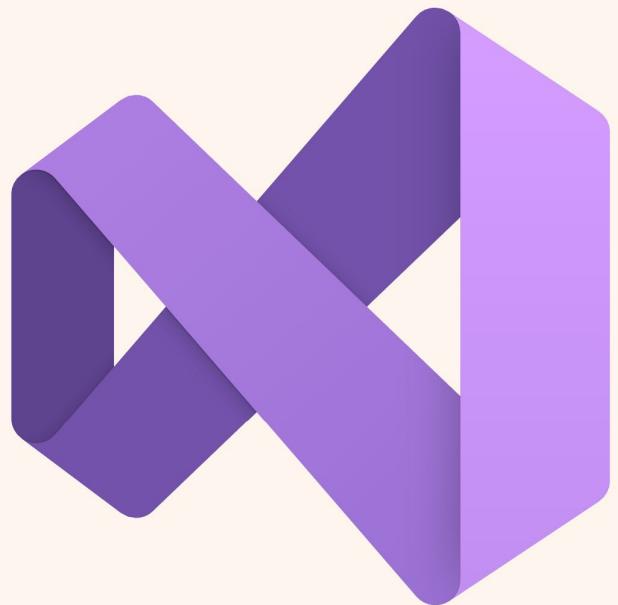
02. **Tunnel** your localhost using a secure tunnel service like Visual Studio dev tunnels, ngrok, etc.



Problems with webhooks

01. Webhooks need to be **publicly accessible** on the **internet**.
How do you develop webhooks on your local machine? 

02. **Malicious actors** can also access webhooks.
How do we secure them? To be continued.



**Develop webhooks locally
with Visual Studio dev tunnels**



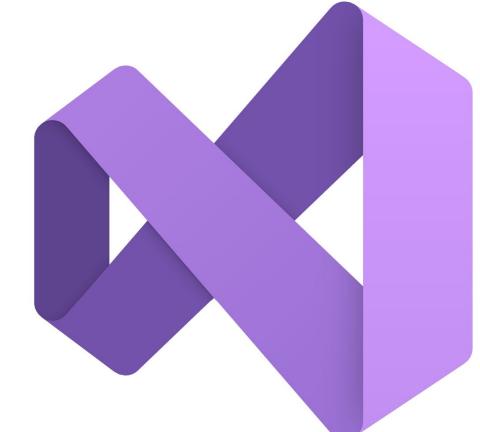
What is dev tunnels

- ✓ Dev tunnels creates a **tunnel from localhost to a public URL**.
HTTP requests send to the public URL are forwarded to your app running on localhost.
- ✓ Only supported in **ASP.NET Core** at the moment.
- ✓ Supports **persistent** and **temporary** tunnels
- ✓ Built-in authentication/authorization



Dev tunnels is a preview feature

- ✓ Available in recent versions of **Visual Studio 2022** and **Visual Studio 2022 Preview** (not VS Code or VS4Mac)
- ✓ VS 2022 (& preview) has **built-in UI** for dev tunnels.



VS

~~VS 2022 uses `launchSettings.json` for dev tunnels,~~
~~and VS 2022 Preview has **built-in UI** for dev tunnels.~~
~~This will be the same for both eventually.~~





Step 0: Open or Create an ASP.NET Core project

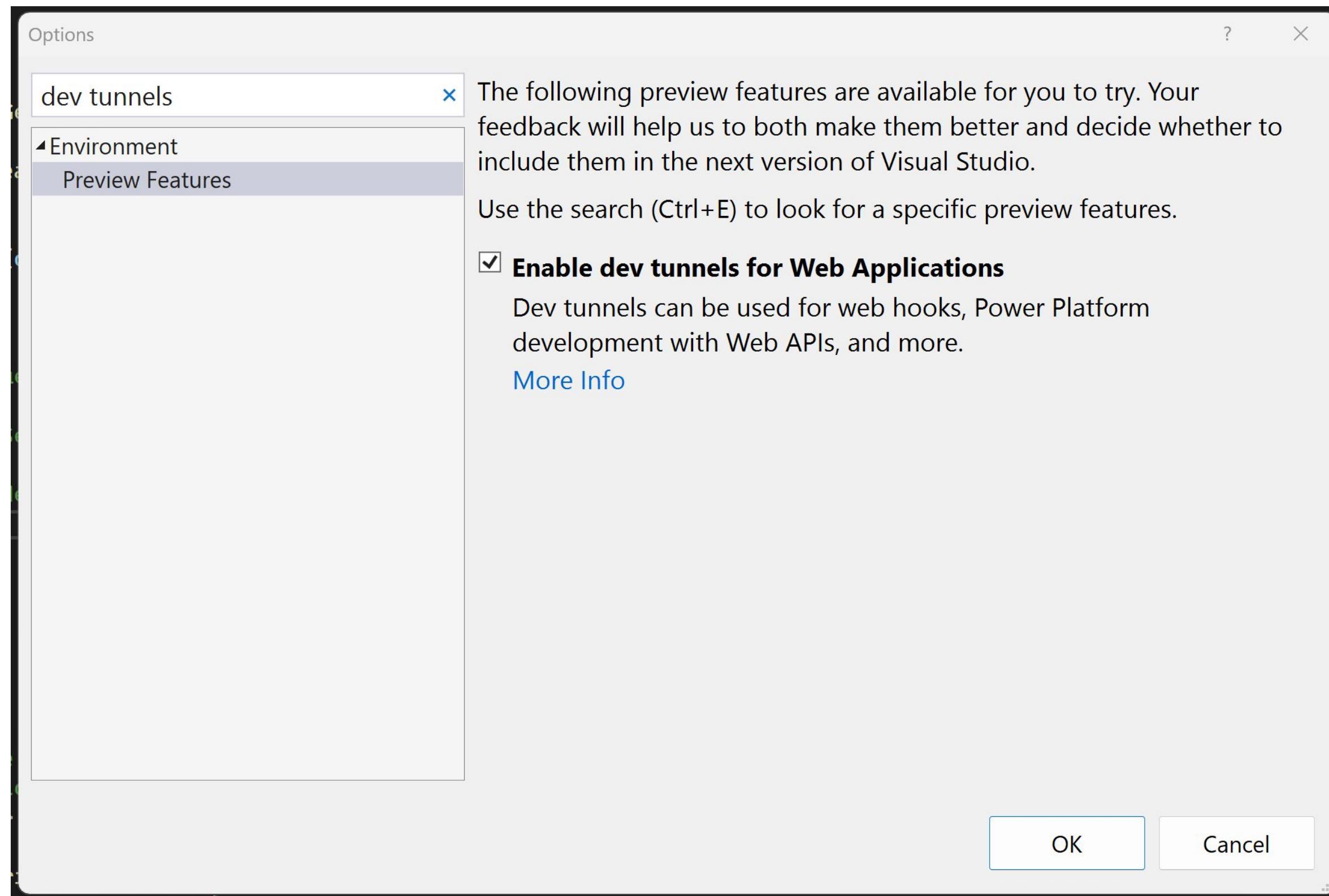
The screenshot shows the 'Create a new project' dialog in Visual Studio. At the top, there's a search bar with 'ASP.NET Core' and a 'Clear all' button. Below it are dropdowns for 'All languages', 'All platforms', and 'All project types'. A descriptive text box explains that the 'ASP.NET Core Empty' template is for creating an ASP.NET Core application with a RESTful HTTP service, and it can also be used for MVC Views and Controllers. It lists supported languages (C#, Web) and platforms (Linux, macOS, Windows, Cloud, Service, WebAPI). Two project templates are listed:

- ASP.NET Core Empty**: An empty project template for creating an ASP.NET Core application. This template does not have any content in it. It supports C# and Web, and runs on Linux, macOS, Windows, Cloud, Service, and Web.
- ASP.NET Core Web App (Model-View-Controller)**: A project template for creating an ASP.NET Core application with example ASP.NET Core MVC Views and Controllers. This template can also be used for RESTful HTTP services. It supports C# and Linux, and runs on macOS, Windows, Cloud, Service, and Web.

A 'Next' button is at the bottom right.

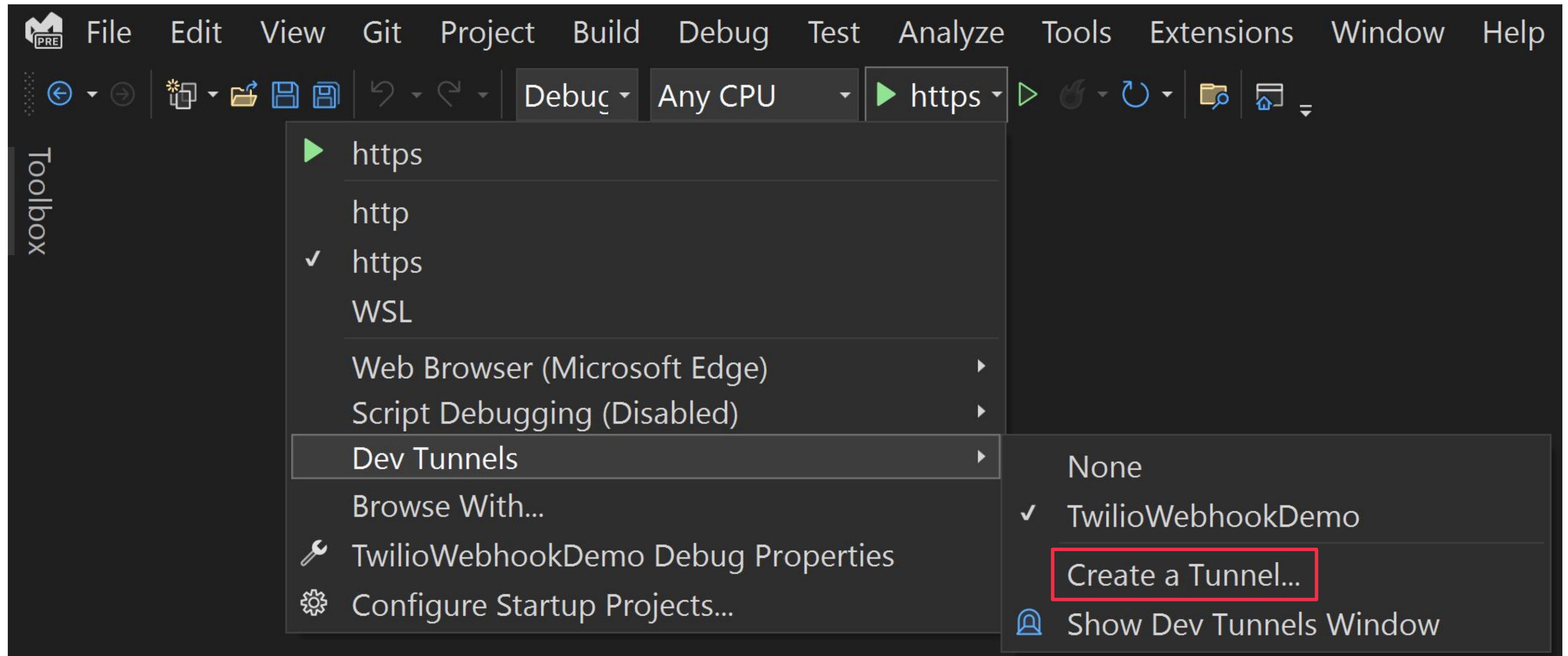


Step 1: Enable dev tunnels preview feature in VS options





Step 2 for VS 2022: Create tunnel with UI





Step 2 for VS 2022: Create tunnel with UI

Account
The account that will be used to create the tunnel
 niels [REDACTED] (Microsoft account) ▾

Name
This name is only used in Visual Studio when selecting a tunnel
TwilioWebhookDemo

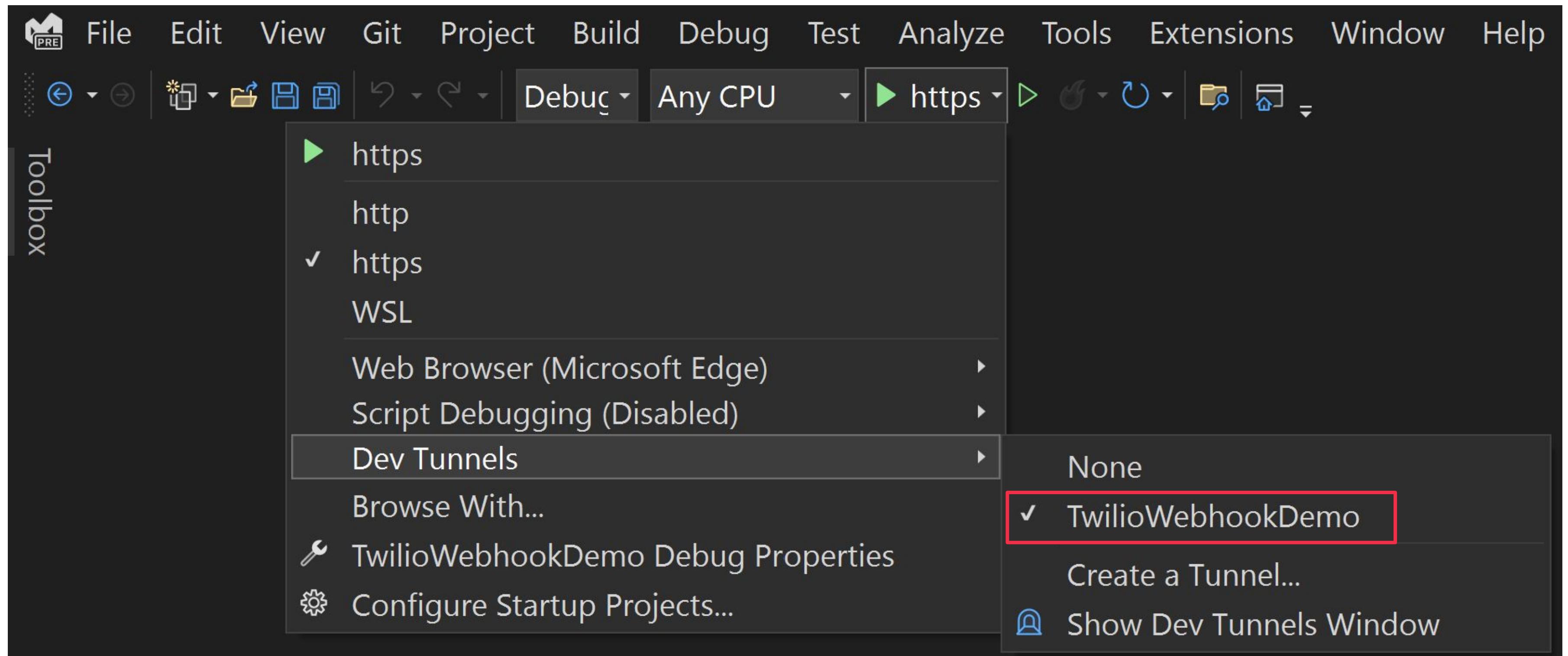
Tunnel Type
This determines the lifetime of the URL. A persistent tunnel will typically reuse the same URL. A temporary tunnel will use a new URL each time Visual Studio is started.
Persistent ▾

Access
This determines who can access the tunnel. When set to private, the tunnel is only accessible to the account which created it. Organizational will only be accessible to the organization(s) the account is a member of. Public does not require any authentication and is not recommended.
Public ▾

OK Cancel

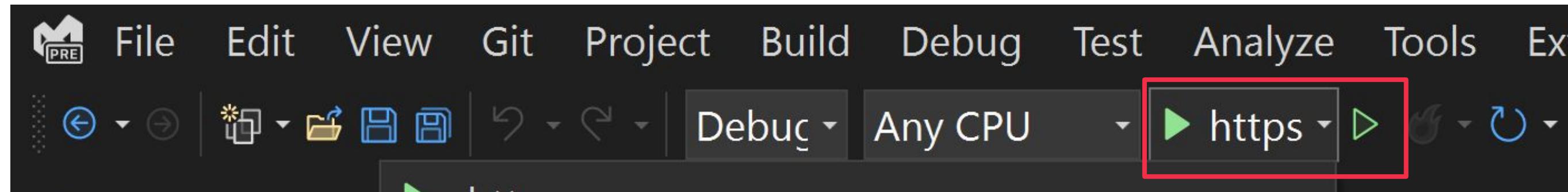


Step 2 for VS 2022: Enable tunnel with UI





Step 3: Start your project as usual





SUCCESS



The screenshot shows a Microsoft Edge browser window with a purple title bar. The address bar displays the URL <https://xvwtvvdp-7165.use.dev tunnels.ms>. The main content area contains a large warning message and a list of bullet points. At the bottom are two buttons: 'CONTINUE' in blue and 'REPORT UNSAFE PAGE' in red.

You are about to connect to a developer tunnel at: **xvwtvvdp-7165.use.dev tunnels.ms**

- Only continue to visit the website if you trust whoever sent you the link
- Personal information, such as credit card numbers or passwords, you disclose may be available to the developer of this site
- We will only show this warning once per tunnel
- This tunnel was created 2 hours ago

[CONTINUE](#) [REPORT UNSAFE PAGE](#)



Demo: Twilio SMS webhook with ASP.NET Core

Minimal API Twilio webhook

— □ ×

```
using Twilio.AspNetCore;
using Twilio.TwiML;

var builder = WebApplication.CreateBuilder(args);

var app = builder.Build();

app.MapPost("/message", () => new MessagingResponse()
    .Message("Ahoy!")
    .ToTwiMLResult()
);

app.Run();
```

Source code: <https://github.com/Swimburger/TwilioWebhookDemo>

©2023 TWILIO INC. ALL RIGHTS RESERVED



Demo: Automatically configure webhook

Configure webhook URL

- □ ×

```
var devTunnelUrl = Environment.GetEnvironmentVariable("VS_TUNNEL_URL");
using var scope = app.Services.CreateScope();
var twilioClient = scope.ServiceProvider.GetRequiredService<ITwilioRestClient>();
await IncomingPhoneNumberResource.UpdateAsync(
    pathSid: app.Configuration["Twilio:PhoneNumberSid"],
    smsUrl: new Uri($"{devTunnelUrl}message", UriKind.Absolute),
    client: twilioClient
);
```



Securing your webhooks

-  Webhooks are publicly accessible, so anyone including malicious actors can access them
-  Every webhook provider supports certain security features you need to implement to protect your webhook server



Securing your Twilio webhook

01. Retrieve **x-twilio-signature** header
02. Combine URL and parameters, and hash using Hash-based Message Authentication Code (HMAC) with your Twilio Auth Token as the secret
03. Securely compare the header and hash



Securing your Twilio webhook

The **Twilio.Security.RequestValidator.Validate** method does this validation for you. This method can be found in the **Twilio** helper library.
(<https://github.com/twilio/twilio-csharp>)

The **Twilio.AspNetCore** library wraps the above method as an MVC filter, Endpoint filter, and Middleware, to make integration easier.
(<https://github.com/twilio-labs/twilio-aspnet>)



Demo: Secure your Twilio webhook

```
Validate Twilio wehook - □ ×

using Microsoft.AspNetCore.HttpOverrides;
using Twilio.AspNet.Core;
using Twilio.TwiML;

var builder = WebApplication.CreateBuilder(args);

builder.Services.AddTwilioRequestValidation();

builder.Services.Configure<ForwardedHeadersOptions>(
    options => options.ForwardedHeaders = ForwardedHeaders.All
);

var app = builder.Build();

app.UseForwardedHeaders();

app.MapPost("/message", () => new MessagingResponse()
    .Message("Ahoy!")
    .ToTwiMLResult()
).ValidateTwilioRequest();

app.Run();
```

Source code: <https://github.com/Swimburger/TwilioWebhookDemo>



Alternatives to VS dev tunnels

- ✓ Not everybody uses **Windows, Visual Studio 2022, and ASP.NET Core**. What other options are there?
- ✓ There are a lot of free and paid tunnel services that work independent of Operating System, IDE, and framework.
- ✓ The most popular tunnel service is **ngrok**, a cross-platform command-line utility.
(Download it at ngrok.com)



Demo: Secure your Twilio webhook

```
> ngrok http http://localhost:5227
```

```
ngrok

Check which logged users are accessing your tunnels in real time https://ngrok.com/s/app-users

Session Status          online
Account                 Niels Swimberghe (Plan: Free)
Update                  update available (version 3.1.1, Ctrl-U to update)
Version                 3.1.0
Region                  United States (us)
Latency                 14ms
Web Interface           http://127.0.0.1:4040
Forwarding              https://5250-71-163-163-158.ngrok.io → http://localhost:5227

Connections             ttl     opn     rt1     rt5     p50     p90
                        0       0      0.00    0.00    0.00    0.00
```



Want to learn more about webhooks?

✓ Check out webhooks.fyi by ngrok. This website has a directory of webhooks, their security controls, and best practices for webhook consumers and providers.

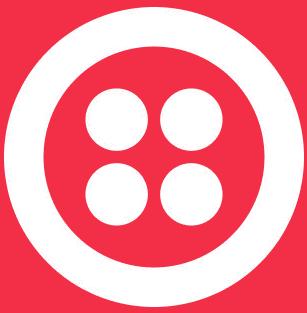
✓ Tutorials:

- Configure Twilio Webhooks automatically with Visual Studio dev tunnels during ASP.NET Core startup (<https://bit.ly/AutoDevTunnelsTwilio>)
- Integrate ngrok into ASP.NET Core startup and automatically update your webhook URLs (<https://bit.ly/AutoNgrokTwilio>)



Want to learn more .NET?

- ✓ .NET tutorials by Twilio: <https://www.twilio.com/blog?tag=net>
- ✓ Tutorials about .NET, Azure, web development, and more
on my personal blog: <https://swimburger.net>



Thank You