Sending Voice Messages with Twilio in .NET

Twilio is a powerful cloud communications platform that allows developers to easily integrate communication services like SMS messaging, voice calls, video conferencing, and email into their applications. By providing developer-friendly APIs, Twilio eliminates the need to build or maintain complex telecom infrastructure. This guide demonstrates how to send voice messages using Twilio in a .NET application, including static and dynamic message handling.

Setting Up Twilio in Your .NET Application

To get started with Twilio, you'll need the following:

- A Twilio account with access to your Account SID and Auth Token.
- A verified Twilio phone number to send messages.
- A hosted TwiML (Twilio Markup Language) file or a dynamically generated TwiML response to dictate the voice message content.

Example: Sending a Voice Message

The following code demonstrates sending a voice message using Twilio's .NET library:

```
using System;
using Twilio;
using Twilio.Rest.Api.V2010.Account;
using Twilio.Types;

class Program
{
  static void Main(string[] args)
  {
    // your Twilio credentials
    const string accountSid = "YourAccountSid";
    const string authToken = "YourAuthToken";

    // initialize the twilio client
    TwilioClient.Init(accountSid, authToken);

    // create the call
    var call = CallResource.Create(
      to: new PhoneNumber("+1234567890"), // the recipient phone number
      from: new PhoneNumber("+0987654321"), // your Twilio phone number
      url: new Uri("http://demo.twilio.com/docs/voice.xml") // URL of TwiML
instructions/message
    );

    Console.WriteLine($"Voice message sent with SID: {call.Sid}");
```

```
  }
}
```

What is TwiML?

TwiML (Twilio Markup Language) is XML-based and dictates what actions Twilio performs during a phone call. For instance, TwiML can specify text-to-speech instructions or audio playback. This Static TwiML example is what is spoken when the call connects.

Example Static TwiML File:

```
<Response>
<Say voice="woman">Thanks for trying our documentation. Enjoy!</Say>
<Play>http://demo.twilio.com/docs/classic.mp3</Play>
</Response>
```

Dynamically Generating TwiML

Dynamic TwiML allows you to create customized voice messages at runtime. You can generate TwiML using an ASP.NET Core API endpoint.

Example API Controller:

```
using Microsoft.AspNetCore.Mvc;
using Twilio.TwiML;

[Route("api/[controller]")]
[ApiController]
public class VoiceController : ControllerBase
{
  [HttpGet("dynamic-message")]
  public IActionResult GetDynamicMessage([FromQuery] string message)
  {
    // Create a TwiML response
    var response = new VoiceResponse();

    // Add a Say verb with the dynamic message
    response.Say(message, voice: "alice");

    // Return TwiML as XML
    return Content(response.ToString(), "application/xml");
  }
}
```

Example C# Code to Call Dynamic Endpoint:

```
var messageToSay = "Hello, this is your dynamic message from Twilio!";
```

```
var call = CallResource.Create(
  to: new PhoneNumber("+1234567890"), // Recipient phone number
  from: new PhoneNumber("+0987654321"), // Your Twilio phone number
  url: new Uri($"https://yourapp.com/api/voice/dynamic-
message?message={Uri.EscapeDataString(messageToSay)}")
);

Console.WriteLine($"Voice message sent with SID: {call.Sid}");
```

Accessing TwiML from External Sources

To host your TwiML file on a server, retrieve it via HTTP, and process it programmatically, you can use an HTTP client to fetch and parse the XML.

Example Implementation:

```
static async System.Threading.Tasks.Task<String> GetXMLVoiceMsgText(string
url)
{
  string retval = String.Empty;

  try
  {
    using (HttpClient client = new HttpClient())
    {
      // download the XML content
      string xmlContent = await client.GetStringAsync(url);

      // parse the XML content using XDocument
      XDocument xDoc = XDocument.Parse(xmlContent);

      retval = xDoc.ToString();
    }
  }
  catch (Exception)
  {
    retval = "error";
  }
  return retval; // return the TwiML content
}
```

Conclusion

Twilio's voice messaging API offers a seamless way to integrate voice capabilities into your application. Whether using static TwiML for predefined messages or dynamically generating messages in real time, Twilio simplifies the process of creating robust voice features. By leveraging Twilio's APIs and its support for TwiML, you can craft engaging voice experiences for users without dealing with the complexities of telecom infrastructure.

http://www.seandrew.info - https://www.linkedin.com/in/seanmdrew