Responding to recipient voice message keypad input (digits)

Twilio is a cloud communications platform that allows developers to integrate various communication services, such as voice, messaging, video, and email, into their applications. Twilio provides APIs that simplify the process of adding features like SMS messaging, phone calls, and video conferencing without needing to build or maintain telecom infrastructure.

To gather user input (keypad digits) during a voice call and respond based on the input, you can use Twilio's `Gather` verb within a TwiML response. Some sample showing how to gather keypad input and provide responses based on that input.

Setup an Endpoint to gather the response

Create an ASP.NET Core API controller with an endpoint that includes the Gather verb to request keypad input.

Example Implementation

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

[Route("api/[controller]")]
[ApiController]
public class VoiceController : ControllerBase
{
    [HttpGet("gather")]
    public IActionResult Gather()
    {
        var response = new VoiceResponse();

        // create a Gather verb to listen for keypad input
        var gather = new Gather(
            input: new List<Gather.InputEnum> { Gather.InputEnum.Dtmf }, // listen for keypad input
            numDigits: 1, // gather 1 digit
            action: new Uri("https://yourapp.com/api/voice/handle-key") // redirect to handle-key endpoint after gathering input
        );

        gather.Say("Press 1 for sales, 2 for support, or 3 for billing."); // add instructions to the Gather verb
        response.Append(gather); // add the Gather verb to the response
        response.Say("We didn't receive any input. Please try again."); // provide a fallback if no input is gathered

        return Content(response.ToString(), "application/xml");
    }
}
```

Handle the Gathered input

Create another endpoint to handle the user's input and respond based on the keypad digit they entered.

Example Implementation

```csharp
[HttpPost("handle-key")]
public IActionResult HandleKey([FromForm] string Digits)
{
  var response = new VoiceResponse();

  // check the gathered digit and respond accordingly
  switch (Digits)
  {
    case "1":
      response.Say("You pressed 1. Connecting you to sales.");
      // Here you could redirect to another call or add more actions
      break;
    case "2":
      response.Say("You pressed 2. Connecting you to support.");
      // Redirect to support, etc.
      break;
    case "3":
      response.Say("You pressed 3. Connecting you to billing.");
      // Redirect to billing, etc.
      break;
    default:
      response.Say("Invalid option. Please try again.");
      response.Redirect(new Uri("https://yourapp.com/api/voice/gather")); // Redirect back to gather if invalid input
      break;
  }

  return Content(response.ToString(), "application/xml");
}
```

<u>Trigger the call and specify the new Gather EndPoint</u>

Example Implementation

```
var call = CallResource.Create(
  to: new PhoneNumber("+1234567890"), // recipient's phone number
  from: new PhoneNumber("+0987654321"), // your Twilio phone number
  url: new Uri("https://yourapp.com/api/voice/gather")
);

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

Complete Flow Overview

<u>Call Initiation:</u> Twilio makes the call and directs it to /api/voice/gather.

<u>Gather Input:</u> The user hears a message and enters a digit.

<u>Handle Key:</u> The digit is captured, and the response is given based on the selection.

<u>Redirection:</u> If needed, invalid input sends the user back to the gather step for retry.