

Gathering Keypad Input in Voice Calls with Twilio and .NET

Twilio, a cloud communications platform, enables developers to integrate communication features such as SMS, voice, video, and email into applications effortlessly. Among its many capabilities is the ability to gather user keypad input (DTMF tones) during a voice call. This guide walks you through the process of implementing a feature that collects keypad input and responds dynamically in a .NET application.

Overview of Twilio's Gather Verb

The Gather verb in TwiML (Twilio Markup Language) is used to capture user input during a call. By using the Gather verb, you can prompt users to press keys on their phone's keypad and respond based on their input.

Step 1: Set Up an Endpoint for Keypad Input

To begin, create an endpoint using ASP.NET Core to serve TwiML that includes the Gather verb. This endpoint provides instructions to the caller and waits for a single digit of input.

Example Implementation

```
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
            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");
    }
}
```

Step 2: Handle Keypad Input

Create another endpoint to handle the user's input and respond based on the keypad digit they entered. Once the user provides input, this endpoint processes the received digit and returns an appropriate response based on their selection.

Example Implementation

```
[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");
}
```

Step 3: Initiate the Call

Trigger the voice call and direct it to the endpoint created for gathering input (/api/voice/gather).

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

Call Initiation: A voice call is initiated, and the user is directed to the /api/voice/gather endpoint.

Gather Input: The user hears a message and provides input by pressing a key on their phone.

Input Handling: The digit entered is sent to the /api/voice/handle-key endpoint, which responds based on the user's choice.

Redirection: If the input is invalid, the user is redirected back to the gather step for another attempt.

Conclusion

Using Twilio's Gather verb in combination with .NET provides a powerful way to collect user input during calls and handle it dynamically. This approach is ideal for creating interactive voice response (IVR) systems, ensuring a seamless user experience for callers. By implementing TwiML endpoints, you can prompt users, process their input, and respond with tailored actions, all while leveraging Twilio's scalable communications platform.