



Programmable Voice

› Getting Started with Programmable Voice

› Tutorials

› API Reference

› TwiML

› Media Streams

Voice Conference

› Conversational Intelligence

▼ ConversationRelay

Overview

Onboarding

Picking a voice

Getting and sending WebSocket messages

Best practices

<ConversationRelay> TwiML noun

› Voice Insights

› Programmable Voice SIP

› Bring Your Own Carrier (BYOC) Trunking

› Client-Side SDKs

› Best Practices &

On this page ▼

Onboarding



Twilio's ConversationRelay extends voice interactions by integrating real-time speech recognition and synthesis into your applications. You can use this service to create flexible and responsive conversational experiences.

This guide walks you through the process of integrating your Twilio application with the ConversationRelay service.

Prerequisites

Complete these prerequisite steps before starting the integration process:

Set up your Twilio account

Ensure you have a Twilio account set up and log into the [Twilio Console](#) .

Configure WebSocket server

Set up a [WebSocket](#) server to handle communication with ConversationRelay.

You can set up a WebSocket server using various frameworks and programming languages, such as [Node.js with Fastify](#) , [Python with FastAPI](#) , or others that support WebSocket protocols.





Programmable Voice

- › Getting Started with Programmable Voice
- › Tutorials
- › API Reference
- › TwiML

Media Streams

Voice Conference

- › Conversational Intelligence

▼ ConversationRelay

Overview

Onboarding

Picking a voice

Getting and sending
WebSocket messages

Best practices

<ConversationRelay>
TwiML noun

- › Voice Insights
- › Programmable Voice SIP
- › Bring Your Own Carrier (BYOC) Trunking
- › Client-Side SDKs
- › Best Practices &

Note: Ensure your WebSocket server is accessible via a `wss://` URL.

WebSocket security with signature validation [↗](#)

following the same verification mechanism used for standard Twilio webhooks.

To validate this signature:

1. Extract the `X-Twilio-Signature` header from the incoming WebSocket connection request.
2. Use your Twilio auth token and the request URL to validate this signature.
3. Only accept connections with valid signatures to prevent spoofed requests.

For detailed implementation guidance, refer to [Twilio's webhook security documentation](#).

Integrate Twilio with ConversationRelay [↗](#)

Once you've completed the prerequisites, follow these steps to integrate your Twilio application with ConversationRelay.

Configuration steps in the Twilio Console [↗](#)

1. Log in to your [Twilio Console](#) [↗](#).
2. Navigate to the **Voice** section, select **General** under **Settings**, and turn on the **Predictive and Generative AI/ML Features Addendum** in order to use ConversationRelay.
3. Navigate to the **Voice** section and select **TwiML Apps** under **Manage**.



Programmable Voice

- › Getting Started with Programmable Voice
- › Tutorials
- › API Reference
- › TwiML
- › Media Streams

Voice Conference

- › Conversational Intelligence

▼ ConversationRelay

Overview

Onboarding

Picking a voice

Getting and sending
WebSocket messages

Best practices

<ConversationRelay>
TwiML noun

- › Voice Insights
- › Programmable Voice SIP
- › Bring Your Own Carrier (BYOC) Trunking
- › Client-Side SDKs
- › Best Practices &

5. In the TwiML App settings, configure the Voice URL to point to your application's endpoint (for example, `https://yourserver.com/voice`) where Twilio can

Configure ConversationRelay in your application



Warning

Ensure your WebSocket server is properly configured to handle incoming messages from ConversationRelay. If not, you may encounter connection errors.

After configuring your TwiML App, update your application to handle ConversationRelay interactions:

- Use the `<Connect><ConversationRelay>` TwiML noun to route calls to the [ConversationRelay service](#).
- Specify the WebSocket URL and any [additional attributes](#) needed for your use case.

Example TwiML:

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <Response>
3   <Connect action="https://myhttpserver.com/co
4     <ConversationRelay url="wss://mywebsockets
5   </Connect>
6 </Response>
```





Programmable Voice

- › Getting Started with Programmable Voice
- › Tutorials
- › API Reference
- › TwiML
- › Media Streams

Voice Conference

- › Conversational Intelligence

▼ ConversationRelay

[Overview](#)[Onboarding](#)[Picking a voice](#)[Getting and sending WebSocket messages](#)[Best practices](#)[<ConversationRelay> TwiML noun](#)

- › Voice Insights
- › Programmable Voice SIP
- › Bring Your Own Carrier (BYOC) Trunking
- › Client-Side SDKs
- › Best Practices &

Develop your application with ConversationRelay

Real-time speech recognition

Use ConversationRelay's real-time speech recognition to process caller input and respond dynamically.

Text-to-speech synthesis

Use ConversationRelay to convert text responses into natural-sounding speech, providing a seamless conversational experience.




For optimal speech quality, apply proper text normalization techniques (such as writing out numbers as words or spelling out abbreviations). See the [ConversationRelay documentation](#) for detailed text normalization best practices.

Custom parameters

Pass [custom parameters](#) to ConversationRelay to tailor interactions based on specific use cases.

Next steps

Learn more about building applications with ConversationRelay by exploring the following tutorials:

- [Integrate OpenAI with Twilio Voice Using ConversationRelay](#) 
- [Building Voice Bots with Twilio's ConversationRelay](#) 
- [ConversationRelay Application and Architecture for Voice AI Applications Built on AWS](#) 





Programmable Voice

- › Getting Started with Programmable Voice
- › Tutorials
- › API Reference
- › TwiML
- › Media Streams

Voice Conference

- › Conversational Intelligence

▼ ConversationRelay

Overview

Onboarding

Picking a voice

Getting and sending
WebSocket messages

Best practices

<ConversationRelay>
TwiML noun

- › Voice Insights
- › Programmable Voice SIP
- › Bring Your Own Carrier (BYOC) Trunking
- › Client-Side SDKs
- › Best Practices & Troubleshooting

We all do sometimes; code is hard. Get help now from our support team , or lean on the wisdom of the crowd by browsing the Twilio tag on Stack Overflow.

[Terms of service](#)

[Privacy Policy](#)

Copyright © 2025 Twilio Inc.

