







- > 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 >

<ConversationRelay> TwiML noun



Legal notice

ConversationRelay, including the <ConversationRelay> TwiML noun and API, uses artificial intelligence or machine learning technologies. By enabling or using any features or functionalities within Programmable Voice that Twilio identifies as using artificial intelligence or machine learning technology, you acknowledge and agree to certain terms. Your use of these features or functionalities is subject to the terms of the Predictive and Generative AI or ML Features <u>Addendum</u> <a>□.

ConversationRelay isn't compliant with the Payment Card Industry (PCI) 2 and doesn't support Voice workflows that are subject to PCI.



Before using ConversationRelay, you need to complete the onboarding steps and agree to the Predictive and Generative AI/ML Features Addendum. See the ConversationRelay Onboarding **Guide** for more details.

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 &

The <ConversationRelay> TwiML noun under the <Connect> verb routes a call to Twilio's ConversationRelay Service, providing advanced Al-powered voice interactions.

system to focus on processing conversational AI logic and sending back responses effectively.

In a typical setup, connects to your Al application through a WebSocket, allowing real-time and event-based interaction. Your application receives transcribed caller speech in structured messages and sends responses as text, which ConversationRelay converts to speech and plays back to the caller. This setup is commonly used for customer service, virtual assistants, and other scenarios that require real-time, Albased voice interactions.

<ConversationRelay> attributes ∅

The <ConversationRelay> noun supports the following attributes:

Attribute name	Description
url	The URL to your Web server (must use ws
welcomeGreeting	The message automate played to the caller at answer the call and establish the WebSoconnection.
welcomeGreetingInterruptible	Specifies if the caller interrupt the
	welcomeGreeting

COOKIE PREFERENCES

Skip to content Skip to navigation

Skip to topbar

Twilio Docs	Attribute name	Description "none", "dtmf",
Programmable Voice		"cnoch" or "ony
> Getting Started with Programmable Voice		und tatoe — non
> Tutorials		The language code (f example, "en-US")
> API Reference		applies to both Speed Text (STT) and Text-to
> TwiML	[language]	Speech (TTS). Setting attribute is equivaler
> Media Streams		setting both (ttsLan and (transcriptionLan
Voice Conference		transcriptionical
> Conversational Intelligence		The default language to use for TTS when t text token messag
✓ ConversationRelay		doesn't specify a lang you set both attribut
Overview	ttsLanguage	one overrides the language attribute
Onboarding		can modify this via th
Picking a voice		ttsLanguage field language message
Getting and sending WebSocket messages		send through the Ser Provider Interface (SI
Best practices		
<conversationrelay> TwiML noun</conversationrelay>	ttsProvider	The provider for TTS. Available choices are "Google", "Amazo
> Voice Insights		and ["ElevenLabs"]
> Programmable Voice SIP		The voice used for TT Options vary based c
> Bring Your Own Carrier (BYOC) Trunking	voice	refer to the <u>Twilio TTS</u> Additional voices are
> Client-Side SDKs		available for ConversationRelay.
> Best Practices &		The least
Skip to content Skip to navigation	Skip to topbar	(B) COOKIE PREFERENCES

Twilio Docs	Attribute name	Description
TWIIIO DOCS		starts. If you set both
Programmable Voice		
> Getting Started with Programmable Voice		transcriptionLan
> Tutorials		message you send the SPI.
> API Reference		
> TwiML	transcriptionProvider	The provider for STT (Speech Recognition) Available choices are
> Media Streams	transcriptionProvider	"Google" and "Deepgram".
Voice Conference		
> Conversational Intelligence		The speech model us STT. Choices vary bas the
 ConversationRelay 	speechModel	transcriptionPrc Refer to the provider
Overview		documentation for an accurate list.
Onboarding		
Picking a voice		Specifies if caller spe interrupt TTS playbac
Getting and sending WebSocket messages		Values can be "none "dtmf", "speech"
Best practices	interruptible	"any". For backwar compatibility, Boolea
<conversationrelay> TwiML noun</conversationrelay>		values are also accep true = "any" and
> Voice Insights		<pre>(false) = ("none").</pre>
> Programmable Voice SIP		Specifies whether the system sends Dual-to
> Bring Your Own Carrier (BYOC) Trunking	dtmfDetection	multi-frequency (<u>DTN</u> keypresses over the WebSocket. Set to t:
> Client-Side SDKs		turn on DTMF events
> Best Practices &	reportInputDuringAgentSpeech	Specifies whether yo
Skip to content Skip to navigation	Skip to topbar	cookie preference

Twilio Docs	Attribute name	Description while the agent is say
		While the agent is spe
Programmable Voice		
 Getting Started with Programmable Voice 		"any" before May,
> Tutorials		and it's now "none"
> API Reference		Specifies if the TTS of current talk cycle can
> TwiML	preemptible	text tokens from the subsequent talk cycle interrupt.
> Media Streams		тенире.
Voice Conference		A comma-separated words or phrases tha
> Conversational Intelligence	hints	Speech-to-Text recog for uncommon word product names, or do
→ ConversationRelay		specific terminology. similarly to the <u>hint</u> attribute in <gather< th=""></gather<>
Overview		<u>attribute in Coatrier</u>
Onboarding		A space-separated lis
Picking a voice		options that you can subscribe to debuggi
Getting and sending WebSocket messages		messages. Options a debugging, speak
Best practices		events, and token played. The debug
<conversationrelay> TwiML noun</conversationrelay>	debug	option provides gene debugging information
> Voice Insights		speaker-events v notify your applicatio
> Programmable Voice SIP		about agentSpeaki and clientSpeaki events. tokens-pla
Bring Your Own Carrier (BYOC) Trunking		will provide message what's just been play
> Client-Side SDKs		TTS.
> Best Practices &	elevenlabsTextNormalization	Specifies whether or apply text normalizat

<u>Skip to content</u> <u>Skip to navigation</u> <u>Skip to topbar</u>



Twilio Docs 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 &



more details.

Nested elements \varnothing

Include nested elements within <ConversationRelay> for more granular configuration. For more information on configuring ConversationRelay, refer to the ConversationRelay Onboarding Guide.

<Language> element ∅

The <Language> element maps a language code to specific TTS and STT settings. Use this element to configure multiple languages for your session.

Example

Connect a Programmable Voice call to Twilio's Con... Node.js 1 const VoiceResponse = require('twilio').twi onse = new VoiceResponse():

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

Skip to content Skip to navigation

- > Client-Side SDKs
- > Best Practices &

```
5 const conversationrelay = connect.conversat
6    url: 'wss://mywebsocketserver.com/webso
7 });
8 conversationrelay language({
```

```
speechModel: 'long'

14 });

15 conversationrelay.language({
16    code: 'en-US',
17    ttsProvider: 'google',
18    voice: 'en-US-Journey-O'
19 });

20

21 console.log(response.toString());
```

Output

Attributes

Attribute name	Description of attributes
code	The language code (for example, "en-US") that applies to both STT and TTS.

ttsProvider

Skip to topbar

The provider for TTS.

Choices a cookie preferences

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 &

Attribute name	Description of attributes
	"Amazon", and
	ttsProvider.
transcriptionProvider	The provider for STT. Choices are "Google" and "Deepgram".
speechModel	The speech model used for STT. Choices vary based on the transcriptionProvider.
language	The language code for the session (for example, "en-US").
customParameter	Custom parameters to be sent in the setup message.

Notes

- If you specify the same language code in both
 <ConversationRelay> and <Language>, the settings in
 <Language> take precedence.
- ConversationRelay provides default settings for commonly used languages.

<Parameter> element 🛷

The <Parameter> element allows you to send custom parameters from the TwiML directly into the initial "setup" message sent over the WebSocket. These parameters appear under the customParameters field in the JSON message.

Example

Skip to topbar

Connect a Programmable Voice call to Twilio's Con... Twilio Docs Node.js Programmable Voice > Getting Started with 3 const response = new VoiceResponse(); Programmable Voice 4 const connect = response.connect(); const conversationrelay = connect.conversatio Tutorials url: 'wss://mywebsocketserver.com/websock 7 {); API Reference 8 conversationrelay.parameter({ name: 'foo', > TwiML value: 'bar' 11 }); Media Streams 12 conversationrelay.parameter({ name: 'hint', Voice Conference value: 'Annoyed customer' 15 }); > Conversational Intelligence 17 console.log(response.toString()); ConversationRelay Overview Output Onboarding Picking a voice 1 <?xml version="1.0" encoding="UTF-8"?> Getting and sending WebSocket messages 2 <Response> <Connect> Best practices <ConversationRelay url="wss://mywebsockets</pre> <Parameter name="foo" value="bar"/> <ConversationRelay> <Parameter name="hint" value="Annoyed cu TwiML noun </ConversationRelay> > Voice Insights </Connect> 9 </Response> > Programmable Voice SIP > Bring Your Own Carrier (BYOC) Trunking **Resulting Setup Message** > Client-Side SDKs

Skip to topbar

setup",

COOKIE PREFERENCES

> Best Practices &

Skip to content Skip to navigation

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 &

```
п. п. п. п.
"customParameters" · {
```

Generating TwiML for <ConversationRelay> 🛷

Connect a Programmable Voice call to Twilio's Con...

```
Node.js
```

```
1 const VoiceResponse = require('twilio').twiml
3 const response = new VoiceResponse();
4 const connect = response.connect({
       action: 'https://myhttpserver.com/connect
6 });
7 connect.conversationRelay({
       url: 'wss://mywebsocketserver.com/websock
       welcomeGreeting: 'Hi! Ask me anything!'
10 });
12 console.log(response.toString());
```

Output

Skip to topbar

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <Response>
    <Connect action="https://myhttpserver.com/
      ConversationRelay url="wss://mywebsocke"
```

Skip to content Skip to navigation

COOKIE PREFERENCES

- > 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 &

- | url | (requirea): The UKL OF your WebSocket Server (must use the wss:// protocol).
- **[welcomeGreeting]** (optional): The message automatically played to the caller after we answer the call and establish the WebSocket connection.

When the TwiML execution is complete, Twilio will make a callback to the action URL with call information and the return parameters from ConversationRelay.

Language settings and their

Language settings refer to configurations for both Text-to-Speech and Speech-to-Text:

- Text-to-Speech (TTS) settings:
 - [ttsLanguage]
 - ttsProvider
 - voice
- Speech-to-Text (STT) settings:
 - transcriptionLanguage
 - transcriptionProvider
 - speechModel

Configure language settings ∅

Configure language settings in two places:

Twilio Docs

- 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 &

- 2. **Within <Language> Elements**: Each <Language> element configures settings for a specific language code. You can include multiple <Language> elements to support multiple
- In <ConversationRelay> , the ttsLanguage | attribute overrides the language | attribute for the default TTS language.
- In <ConversationRelay>, the transcriptionLanguage
 attribute overrides the language attribute for the STT
 language.
- If a <Language> element specifies the same code
 attribute as in <ConversationRelay>, the <Language>
 element's settings take precedence.
- The system uses default values when you don't provide specific settings.

Default Values

- language: Defaults to en-US if not specified.
- **[ttsProvider**]: Defaults to **ElevenLabs** if not specified.
- **transcriptionProvider**: Defaults to **Google** if not specified.
- If you set the ttsProvider attribute without the voice attribute, the system uses a default voice for that provider.
- If you set the transcriptionProvider attribute without the speechModel attribute, the system uses a default model for that provider.
- If you set the voice attribute without the ttsProvider attribute, the system infers the provider from the default or specified ttsProvider.
- If you set the speechModel attribute without the transcriptionProvider attribute, the system infers the provider from the default or specified
 transcriptionProvider.





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 &

For Speech-to-Text (STT) settings:

• At session start, the service uses the

speechModel attributes is invalid, the call disconnects, and the system reports an error in the action callback and error notifications.

• You can change the transcriptionLanguage attribute during the session via the language message you send through the Service Provider Interface (SPI).

For Text-to-Speech (TTS) settings:

- When the lang property is present in the text token message from the SPI, the service uses it to select the TTS voice.
- If the combination of the ttsProvider and voice attributes is invalid, the system sends an error message over the SPI.
- If you don't specify the lang property in the text token, the service uses the current TTS language settings.

Result of TwiML execution Ø

<Connect> action URL callback ∅

When an action URL is specified in the <Connect> verb, ConversationRelay will make a request to that URL when the <Connect> verb ends. The request includes call information and session details.

Example Payloads

Session ended by application example 🤣



Twilio Docs 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 &

```
"CallStatus" · "in-nrngress"
  "SessionStatus": "ended",
  "SessionDuration": "25",
  "HandoffData": "{\"reason\": \"The caller r
13 }
```

Error occurred during session example Ø

```
"CallStatus": "in-progress",
   "From": "client:caller",
   "To": "test:conversationrelay",
   "Direction": "inbound",
   "SessionStatus": "failed",
   "SessionDuration": "10",
   "ErrorCode": "39001",
   "ErrorMessage": "Network connection to WebS
14 }
```

Session completed normally (caller hung up) example @

```
1 {
  "CallStatus": "completed",
  "From": "client:caller",
  "To": "test:conversationrelay",
```

on": "inbound",



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 &

Al nutrition facts @

ConversationRelay, including the ConversationRelay
TwiML nouns and APIs, use artificial intelligence or machine learning technologies.

Our AI Nutrition Facts for ConversationRelay provide an overview of the AI feature you're using, so you can better understand how the AI is working with your data. The below AI Nutrition Label details the ConversationRelay AI qualities. For more information and the glossary regarding the AI Nutrition Facts Label, refer to our AI Nutrition Facts page

Deepgram AI nutrition facts @

Al Nutrition Facts

ConversationRelay (STT and TTS) - Programmable Voice - Deepgram

Description

Generate speech to text in real-time through a WebSocket API in Programmable Voice.

Privacy Ladder LevelN/AFeature is OptionalYesModel TypeAutomatic Speech RecognitionBase ModelDeepgram Nova 2

Trust Ingredients

Base Model Trained with Customer Data

No

Skip to content Skip to navigation Skip to topbar



ConversationRelay uses the Default Base Model provided by the Model Vendor. The Base Model is not trained using Customer Data.

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 &

Training Data Anonymized

N/A

Base Model is not trained using any Customer Data.

Data Deletion

N/A

Customer Data is not stored or retained in the Base Model

Human in the Loop

Yes

Customer can view and listen to the input and output in the customer's own terminal.

Data Retention

N/A

Compliance

Logging & Auditing

Yes

Customer can view and listen to the input and output in the customer's own terminal.

Guardrails

Yes

Customer can view and listen to the input and output in the customer's own terminal.

Input/Output Consistency

Yes

Customer is responsible for human review.

Other Resources

Learn more about this label at nutrition-facts.ai

Learn more about this label at **nutrition-facts.ai**

Google AI nutrition facts ∅

Al Nutrition Facts

ConversationRelay (STT and TTS) -Programmable Voice - Google Al

Description

Generate speech to text in real-time and convert text into natural-sounding speech through a WebSocket API in Programmable Voice.

Skip to content Skip to navigation Skip to topbar

r Level



COOKIE PREFERENCES

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 &

Feature is Optional

/es

Model Generative and Predictive - Automatic Speech **Type** Recognition and Text-to-Speech

Base Model Trained with Customer Data

No

ConversationRelay uses the Default Base Model provided by the Model Vendor. The Base Model is not trained using Customer Data.

Customer Data is Shared with Model Vendor

No

ConversationRelay uses the Default Base Model provided by the Model Vendor. The Base Model is not trained using Customer Data.

Training Data Anonymized

N/A

Base Model is not trained using any Customer Data.

Data Deletion

N/A

Customer Data is not stored or retained in the Base Model

Human in the Loop

Yes

Customer can view and listen to the input and output in the customer's own terminal.

Data Retention

N/A

Compliance

Logging & Auditing

Yes

Customer can view and listen to the input and output in the customer's own terminal.

Guardrails

Yes

Customer can view and listen to the input and output in the customer's own terminal.

Input/Output Consistency

Yes

Customer is responsible for human review.

Other Resources

Learn more about this label at nutrition-facts.ai

Learn more about this label at **nutrition-facts.ai**

Amazon Al nutrition facts 🛷



- > 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 &

Mutritian Eacto

Description

Convert text into natural sounding speech through a websocket API in Programmable Voice.

Privacy Ladder Level N/A **Feature is Optional** Yes **Model Type** Generative and Predictive **Base Model** Amazon Polly Text-to-Speech

Trust Ingredients

Base Model Trained with Customer Data

No

ConversationRelay uses the Default Base Model provided by the Model Vendor. The Base Model is not trained using Customer Data.

Customer Data is Shared with Model Vendor

No

ConversationRelay uses the Default Base Model provided by the Model Vendor. The Base Model is not trained using Customer Data.

Training Data Anonymized

N/A

Base Model is not trained using any Customer Data.

Data Deletion

N/A

Customer Data is not stored or retained in the Base Model.

Human in the Loop

Yes

Customer can view and listen to the input and output in the customer's own terminal.

Data Retention

N/A

Compliance

Logging & Auditing

Yes

Customer can view and listen to the input and output in the customer's own terminal.

Guardrails

Yes

Customer can view and listen to the input and output in the customer's own terminal.

Input/Output Consistency

Yes

Customer is responsible for human review.

Learn more about this label at **nutrition-facts.ai** [7

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 &

AI Nutrition Facts

ConversationRelay (STT and TTS) -Programmable Voice - ElevenLabs

Description

Convert text into a human-sounding voice using speech synthesis technology from ElevenLabs.

Privacy Ladder Level

N/A

Feature is Optional

Yes

Model Type

Predictive

Rase

ElevenLabs Text-To-Speech: Flash 2 and

Model Flash 2.5

Trust Ingredients

Base Model Trained with Customer Data

No

The Base Model is not trained using any Customer Data.

Customer Data is Shared with Model Vendor

No

Programmable Voice uses the default Base Model provided by the Model Vendor. The Base Model is not trained using customer data.

Training Data Anonymized

N/A

Base Model is not trained using any Customer Data.

Data Deletion

N/A

The Base Model is not trained using any Customer Data.

Human in the Loop

Yes

Customers can view text input and listen to the audio output.

Data Customer can review TwiML logs, including Retention < Say > Logs, to debug and troubleshoot for up to 30 days.

Compliance

uditing

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

Customers can view text input and listen to the audio output.

Guardrails

Yes

Other Resources

Learn more about this label at nutrition-facts.ai

Learn more about this label at **nutrition-facts.ai**

Need some help?

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.

