



- Getting Started with Programmable Voice

➤ API Reference

➤ Media Streams

- › Conversational Intelligence

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 &

Getting and sending WebSocket messages



Twilio communicates with your ConversationRelay application using a WebSocket connection. Your application will receive messages from Twilio with the caller's speech and other call events. You'll control your side of the conversation by sending messages back to Twilio with text tokens, media playback requests, DTMF digits, and other commands.

Getting messages from Twilio



Your WebSocket server must validate incoming messages from Twilio using the `X-Twilio-Signature` header. For detailed guidance on setting up signature validation, see [Configure your WebSocket server](#). The messages that you receive from Twilio will be one of the following types:


Setup message



ConversationRelay sends this message immediately after establishing the WebSocket connection.

```
1 {  
2   "type": "setup",  
3   "sessionId": "VX00000000000000000000000000000000"  
4   "id": "CA00000000000000000000000000000000"  
5   "url": "http://localhost:8080/vx/setup?sessionId=VX00000000000000000000000000000000&id=CA00000000000000000000000000000000&url=http%3A%2F%2Flocalhost%3A8080%2Fvx%2Fsetup%3FsessionId%3DVX00000000000000000000000000000000&token="+14151234567",
```

[Skip to topbar](#)

 COOKIE PREFERENCES

 COOKIE PREFERENCES

[Skip to content](#) [Skip to navigation](#) [Skip to topbar](#)



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 &

```
6  "to": "+18881234567",
7  "direction": "inbound",
8  "...": "...",
9  "customParameters" : {
```

Prompt message [↗](#)

ConversationRelay sends this message when the caller says something.

```
1  {
2    "type": "prompt",
3    "voicePrompt": "Hi! Can you tell me about li
4    "lang": "en-US",
5    "last": true
6  }
```

DTMF message [↗](#)

ConversationRelay sends this message when you turn on DTMF detection and the caller presses a key.

```
1  {
2    "type": "dtmf",
3    "digit": "1"
4  }
```

Interrupt message [↗](#)

ConversationRelay sends this message when the caller interrupts TTS playback by speaking.



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 &

```
1 {
2   "type": "interrupt",
3   "utteranceUntilInterrupt": "Life is a comple
4   "durationUntilTermination": "400"
```

Error message [↗](#)

ConversationRelay sends this message when an error occurs during the session.

```
1 {
2   "type": "error",
3   "description": "Invalid message received: {
4 }
```

Sending messages to Twilio [↗](#)

ConversationRelay validates the incoming messages that your application sends us. If validation fails then Twilio returns error [64107](#) with details about the validation failure. A failed validation check does not end the session but the message will not be processed.

Text tokens message [↗](#)

Send text tokens messages to Twilio and ConversationRelay will convert them to speech. We recommend streaming text tokens as soon as they are available, rather than waiting for the entire message to be ready. This reduces latency. Use the `last` attribute to indicate the final token in the talk cycle.

```
1 {
  "text",
  "Hello world!",
```



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 &

```
4  "last": false,  
5  "interruptible": false,  
6  "preemptible": false  
7  }
```

Attribute	Description	validation rules
<code>type</code>	The type of message you're sending to Twilio.	Required. Must equal <code>text</code> .
<code>token</code>	The text that will be converted to speech and played to the caller. Don't trim LLM tokens; the tokens should have appropriate spaces between them.	Required. Cannot be <code>null</code> .
<code>last</code>	Whether this is the final token in the current message. Defaults to <code>false</code> .	Optional, can be <code>true</code> or <code>false</code> .
<code>lang</code>	The language of the text.	Optional, must be one of the supported languages if included.
<code>interruptible</code>	Whether the caller can interrupt this text. Overrides the <code>interruptible</code> attribute in TwiML .	Optional boolean value.
<code>preemptible</code>	Whether subsequent <code>text</code> or <code>play</code> messages will stop this media playback. Overrides	Optional boolean value.



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

Description

Validation rules

the `preemptible`

Request to play media to the caller.

```
1 {
2   "type": "play",
3   "source": "https://api.twilio.com/cowbell.mp3",
4   "loop": 1,
5   "preemptible": false,
6   "interruptible": true
7 }
```

Attribute

Description

Validation rules

`type`

The type of message you're sending to Twilio.

Required. Must equal `play`.

`source`

The URL of the media to play.

Required. Must contain a valid URL.

`loop`

Number of times to play the media. A value of `0` plays it 1,000 times (maximum). Default is `1`.

Optional, integer value.

`preemptible`

If set to `true`, subsequent `text` or `play` messages will stop this media

Optional boolean value





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

Description

Validation rules

playback. Default is

`interruptible`

can interrupt this media. Overrides [the `interruptible` attribute in TwiML](#).

Optional boolean value

Send digits message [↗](#)

Request to send DTMF digits to the caller. ConversationRelay sends digits as per [Twilio's <Play>](#), [digits](#) [attribute](#).

```
1 {  
2   "type": "sendDigits",  
3   "digits": "9www4085551212"  
4 }
```

Attribute

Description

Validation rules

`type`

The type of message you're sending to Twilio.

Required. Must equal `sendDigits`.

`digits`

The DTMF digits to send.

Required. Cannot be `null` or empty. Must only contain the characters 0-9, w, #, and *.

Switch language message [↗](#)

Change the transcription and TTS language during the session. This affects future TTS and STT sessions.





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 &

```
1 {  
2   "type": "language",
```

Attribute	Description	Validation
<code>type</code>	The type of message you're sending to Twilio.	Required. Must be <code>language</code> .
<code>ttsLanguage</code>	The language to use for text-to-speech conversion.	Optional, but must be present. Must be one of the supported languages.
<code>transcriptionLanguage</code>	The language to use for speech-to-text transcription.	Optional, but must be present. Must be one of the supported languages.

End session message [↗](#)

End the session and return control of the call to Twilio through ConversationRelay.

```
1 {  
2   "type": "end",  
3   "handoffData": "{\"reasonCode\":\"live-agent\"",  
4 }
```



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

Description

Validation rules

`handoffData`

A string containing data to pass back in the action callback.

Optional.

Handling errors [↗](#)

For errors, such as messages that ConversationRelay doesn't understand, we will respond with an error message.

If your WebSocket sends unidentified messages to ConversationRelay and the last 10 messages remain unidentified, we will terminate the connection. The status code will be 1007 with the reason "Too many consecutive malformed messages." In that case, we will report an [error 64105](#) "WebSocket Ended."

Limitations [↗](#)

If the WebSocket disconnects unexpectedly in ConversationRelay, we don't reconnect, and the call disconnects with a `failed` status.

WebSocket reconnection logic [↗](#)

In the event of a WebSocket connection error in ConversationRelay, implement reconnection logic by initiating a new `<Connect><ConversationRelay>` request:

- **Re-establish the connection:** If you lose the WebSocket

le the disconnect in your





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


element's `action` URL callback by returning new TwiML containing `<Connect><ConversationRelay>` to restore the session.

This approach helps maintain session stability and consistency following any connection disruptions.

See also

- [ConversationRelay overview](#)
- [TwiML ConversationRelay noun reference](#)

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.

