

Troubleshoot WhatsApp Calling Errors - Reference Guide | Developer Documentation

Troubleshoot WhatsApp Calling Errors - Reference Guide

Updated: Dec 3, 2025

Standard error response

When you receive an error, in the majority of cases the error shape will look like this:
Use the Calling API error codes list below to identify and resolve calling errors.

Calling logs

The **Call Logs** tab in [WhatsApp Manager](#) provides businesses and partners with a detailed, self-service view of call events to aid in call troubleshooting.

The tab displays a table of recent call logs for your business phone numbers. Each call has a log. Each log can have multiple events representing a Graph API request made by the business or a webhook sent by Meta. Each row represents a call, with different columns highlighted to provide information about each call log.

How to view call logs

Navigate to [WhatsApp Manager > Account tools > Phone numbers](#) Select the desired phone

number to view the call logs

Call logs and events

Field	Description
Timestamp	Timestamp of when the call occurred.
Call Direction	Outbound (Business-initiated) or Inbound (User-initiated).
Signaling	Signaling Protocol used to establish the call (SIP, GRAPH_API).
Call ID	WhatsApp call identifier. Note: Please provide this ID when requesting support.
Request ID	Identifier for the request that initiated the call.
Call Details	Additional information containing a log of events during the lifecycle of the call.

Calling error codes

Code	Description	Possible Solutions	HTTP Status Code
100 Invalid Parameter	The Graph API call you are making has an invalid parameter.	Exact error details will be listed in the <code>error_data</code> section of the response payload. If it's an SDP validation related error then the exact issue will be included in the details.	400 Bad request
613 Fetch Call Permission For Consumer Phone Number API Limit Hit	Limit reached for requests to the fetch call permission status API.	Try again later or reduce the frequency or amount of requests to the API the app is making.	429 Too many requests
131009	Interactive Message type, <code>voice_call</code> not supported. Supported types [<code>button</code> , <code>list</code>]	Ensure the sender is in one of the supported countries .	400 Bad request
131030	Recipient phone number is not in allowed list. The error is thrown for both attempted calls and call permission requests. Only occurs when using public test numbers (PTNs).	Add a recipient phone number to recipient list and try again.	400 Bad request

Code	Description	Possible Solutions	HTTP Status Code
138000 Calling not enabled	Calling APIs are not enabled for this phone number	Calling is not enabled on the business phone number. Configure call settings to enable Calling API features	401 Unauthorized
138001 Receiver Uncallable	Receiver is unable to receive calls Reasons can include: The recipient phone number is not a WhatsApp phone number.The recipient has not accepted our new Terms of Service and Privacy Policy.Recipient using an unsupported client. The currently supported clients are Android, iOS, SMB Android and SMB iOS	Confirm with the recipient that they agree to be contacted by you over WhatsApp and are using the latest version of WhatsApp.	400 Bad Request
138002 Concurrent Calls limit	Limit reached for maximum concurrent calls (1000) for the given number	Try again later or reduce the frequency or amount of API calls the app is making.	429 Too many requests
138003 Duplicate call	A call is already ongoing with the receiver	Try again later when the current call ends.	400 Bad request
138004 Connection error	Error while connecting the call	Try again later or investigate the connection params provided to the API.	500 Server error
138005 Call rate limit exceeded	Limit reached for maximum calls that can be initiated by the business phone number	Try again later or reduce the frequency or amount of API calls the app is making.	429 Too many requests
138006 No approved call permission found	No approved call permission from the recipient	Ensure a call permission has been accepted by the consumer	401 Unauthorized
138007 Connect Timeout error	Call was unable to connect due to a timeout	Business did not apply the offer/answer SDP from Cloud API in time. Connect API was not invoked with the answer SDP in time	500 Server error
138009 Call Permission Request Limit Hit	Limit reached for call permission request sends for the given business and consumer pair	When a business sends more than the limit of call permission requests per time period, Call Permission Requests are rate limited. A connected call with a consumer will reset the limits.	400 Bad request

Code	Description	Possible Solutions	HTTP Status Code
138012 Business Initiated Calls Limit Hit	Limit reached for maximum business initiated calls allowed in 24 hours. Currently 100 connected business initiated calls are allowed within 24 hours.	Exact error details will be listed in the <code>error_data</code> section of the response payload. Details will include a timestamp when the next call is allowed.	400 Bad request
138013 Business-initiated calling is not available	Business-initiated calling is not available for this business account phone number.	Confirm that Business Initiated calling is available for this business account phone number in the availability section here .	400 Bad request
138014 Calling is temporarily disabled	Calling APIs for this WhatsApp Business Number have been temporarily disabled for having low quality.	Ensure that your outreach to users is valuable and not spam. Retry the action after the account restrictions are lifted.	400 Bad request
138015 Calling cannot be enabled	Calling APIs cannot be enabled for this phone number.	Check and make sure messaging limit on your phone number is 2000 or more	400 Bad request
138017 Call permission request can't be sent because a permanent permission already exists.	Call permission request cannot be sent because a permanent permission has already been approved by the user for this business account phone number.	No need to send call permission requests.	400 Bad request
138018 WhatsApp Business calling cannot be enabled because technical pre-requisites are not met.	See prerequisites for more details	Configure SIP or ensure there is at-least one app subscribed to the WhatsApp Business Account that also has subscription to the <code>calls</code> webhook field.	400 Bad request
138019 Call setup failed	The WhatsApp client failed to set up the call. This error is sent in the call terminate webhook.	Try again later.	400 Bad request
138020 Relay connection failed	The WhatsApp client failed to establish a connection with the relay server. This error is sent in the call terminate webhook.	Try again later.	400 Bad request

Code	Description	Possible Solutions	HTTP Status Code
138021 Media receive timeout	The WhatsApp client terminated the call due to not receiving any media for a long time. This error is sent in the call terminate webhook.	Confirm that call media is being sent to the relay and try again.	400 Bad request
138022 Media transmit timeout	The WhatsApp client terminated the call due to not transmitting any media for a long time. This error is sent in the call terminate webhook.	Try again later.	400 Bad request
138023 Call accepted but terminated with no media signals	The call was accepted but terminated with no media connection signals. Cloud API could not determine if the call was connected. This error is sent in the call terminate webhook.	Try again later.	400 Bad request
131044	An error webhook is sent on user initiated calls when there is no valid payment method attached	This is similar to the “Business eligibility payment issue” error in messaging.	400 Bad request
131055	Method not allowed. Graph API calls are not allowed for SIP enabled numbers	Use SIP for SIP enabled business numbers.	400 Bad request

SIP errors

Also refer to [SIP specific FAQ](#) for additional troubleshooting info

Business initiated calls

SIP status and message	Description	Possible Solutions
400 Asset not found, invalid business phone number	The phone number in the <code>From</code> header of the INVITE is invalid and does not correspond to a registered account on WhatsApp Business Platform	Check the phone number and resend the INVITE with the correct <code>From</code> header value. See examples

SIP status and message	Description	Possible Solutions
<p>403</p> <p>SIP server <code>foo.com</code> from INVITE does not match any SIP server configured for phone number id [ID]</p>	<p>Your SIP INVITE to Meta has 'foo.com' in the <code>From</code> header (for example, <code>15555551234@foo.com</code>). But there is no SIP server with that hostname configured on the business phone number.</p>	<p>Check your SIP configuration and ensure it matches the domain used in the <code>From</code> header. The <code>hostname</code> in the SIP configuration should match the domain in the <code>From</code> header of your SIP INVITE. It is also valid for the SIP configuration <code>hostname</code> to be a sub-domain of the domain in <code>From</code> header. For example, SIP server configured is <code>bar.foo.com</code> and the From header is <code>15555551234@foo.com</code></p>
<p>403</p> <p>No Approved Call Permission Found: No approved call permission from the recipient</p>	<p>There is no WhatsApp user with specified number or user has not accepted our new Terms of Service or there is no permission from user to call the business. Meta can't reveal if a phone number is on WhatsApp with certainty, due to privacy reasons. So this is the same error you'll receive for multiple reasons.</p>	<p>Double check the user phone number and make sure you obtain the user permission. See Obtain User Call Permissions for more information</p>
<p>403</p> <p>The app [APP_ID] configured for SIP server <code>example.com</code> is not authorized for phone number id [ID]</p>	<p>When you configure SIP server on a business phone, Meta extracts the app id from the access token and stores a mapping from the SIP server to that app id. When you send a SIP INVITE, Meta consults that mapping to know which app is making the SIP request. This error means that app does not have <code>whatsapp_business_messaging</code> permission on the business phone.</p>	<p>Check that you are using the right app and make sure it has permissions on the business phone. If you're able to send message using the app's access token, that means the app has the right permission. Otherwise you may need to delete and add your SIP server with the correct app and retry your SIP INVITE. See Configuring SIP settings.</p>
<p>403</p> <p>Business Initiated Connected Call Per Day Limit Hit</p>	<p>Limit reached for maximum business initiated calls allowed in 24 hours. Currently 5 connected business initiated calls are allowed within 24 hours.</p>	<p>See more info about the limits and adjust your calling rate accordingly</p>
<p>403</p> <p>Additional reasons for 403 include Duplicate call because there can be only one active between a given business and user accountInvalid digest auth responseBusiness eligibility payment issueSDP Validation Error</p>	<p>Read each error for details</p>	<p>Consult earlier sections in this page like Calling error codes</p>
<p>404</p> <p>Not found</p>	<p>SIP INVITEs using IP in request URI are not allowed</p>	<p>Use the correct request URI. See examples</p>

SIP status and message	Description	Possible Solutions
407 Proxy Authentication Required	Meta mandates digest authentication for your SIP INVITEs and this is the standard challenge response from Meta	Resend the SIP INVITE with digest response. See examples
408 RTP Timeout	The WhatsApp client terminated the call due to not receiving any media for a long time.	See Media related issues
480 Temporarily Unavailable	WhatsApp user is not reachable or did not answer the call	Try messaging or calling later. Unanswered calls impact call permissions. Learn more
486 User declined the call	WhatsApp user declined your call	Try messaging or calling later. Rejected calls impact call permissions. Learn more
487 Request Terminated	Business canceled their SIP INVITE using SIP CANCEL	Expected return for your SIP INVITE when you CANCEL the INVITE before Meta response to your INVITE
503 Service Unavailable	Generic internal error.	Retry your request after some time or consult Meta support

Media related issues

Issue	Description	Possible Solutions
Call disconnects after 20 seconds into the call	At the start of the call, if there is no media flowing from business to WhatsApp user for 20 seconds, WhatsApp client will disconnect the call due to no media	Check your media server to make sure it's initiating the media session and sending media packets Check your firewall to rule out any packet drops at your edge Capture network traffic using pcap or a similar tool and attach it to the support ticket for further troubleshooting by Meta
Can't hear audio and call disconnects after 30 seconds	After the call is connected and established, if there is no media packet from business to WhatsApp user for 30 seconds, WhatsApp client will disconnect the call due to no media reception	Make sure you're at-least sending RTCP packets even when there is silence or when you are waiting for user input (for example, IVR) Check your media server to understand why it stops sending media packets to Meta Capture network traffic using pcap or a similar tool and attach it to the support ticket for further troubleshooting by Meta

Audio clipping issue and solution

When bridging the WhatsApp Consumer media leg via WebRTC to another media leg, such as [SIP](#), it's important to be aware of potential audio clipping issues and how to prevent them. A common symptom of this issue is that the WhatsApp consumer may miss approximately one second of audio from the business. For instance, if an IVR system is playing a sequence like "1-2-3," the consumer might only hear "2" or "3." The extent of audio loss varies based on the specific integration and the severity of the issue.

The sequence diagram below illustrates this problem using an example where the WebRTC leg from the Cloud API is bridged with a SIP media leg. Depending on the SIP vendor and implementation, the media from the SIP User Agent might start playing at step 11, but it won't reach the WhatsApp consumer until step 18.

Sequence diagram

Sequence diagram showing WebRTC to SIP call bridging flow (*Right click image and choose "Open in new tab" for enlarged image*)

Components of the sequence diagram

WAConsumer is a WhatsApp user calling the business phone number using WhatsApp mobile app
Meta is the Cloud API product
MetaWebrtcEndpoint is the WebRTC agent on the Meta infrastructure
BizIntegration is the webhook server receiving calls related webhooks and the app server with business logic to invoke Cloud API Graph APIs
BizWebrtcEndpoint is the WebRTC termination point as well as the SIP UAC typically in a media server on your side.
BizSipEndpoint is the SIP User Agent (UA) often representing IVR or Business Agent.

Key points to note

Upon receiving the connect webhook (step 5 above), initialize your WebRTC agent, prepare the SDP answer, and make the Pre-accept Graph API call with the SDP answer. This pre-establishes the WebRTC connection in anticipation of an 'accept/ok' response from the SIP User Agent (UA). In most cases, the consumer's call is answered, such as by an automated recording or IVR system. At step 11 above, wait for the ICE process to complete the creation of valid lists, indicating the near completion of ICE connection establishment (RFC reference). If the SIP UA rejects the call instead of returning a 200 OK, use the terminate Graph API to end the call. If the consumer connection is ready before the SIP connection, the consumer may experience a few milliseconds of silence, which is preferable to losing some initial audio from the business.

Root cause

When bridging two media legs, it's crucial for the media server to synchronize their readiness to prevent audio clipping. In this scenario, the *BizWebrtcEndpoint* is allowing the SIP User Agent (UA)

to transmit media significantly earlier than the WebRTC leg is prepared to receive it. As a result, packets are being dropped at the media server, leading to audio clipping issues.

Suggested solutions

Below is a high-level summary of alternative approaches to address audio clipping, intended to stimulate discussion and consideration. While each option has its drawbacks, you can assess their feasibility in the context of your specific requirements.

Use SDES: [Configure SDES](#) on your business number instead of DTLS. A common reason for delayed setup of the media leg between your infrastructure and Meta is DTLS handshake completion. Meta cannot complete DTLS handshake until it receives your SDP. You may not be able share your SDP with Meta until your internal endpoint responds to your with their SDP. Common DTLS implementations have a retry interval of 1s, 2s, 4s, and so on. After you send Meta the SDP, there is often a ~3 second delay before we receive your DTLS client hello packet and this is when your internal endpoint is sending media but it is dropped. When you switch to SDES you can directly send SRTP after you send us your SDP.**Delayed Audio Playback:** Instruct the SIP User Agent (UA) to wait for an ACK from BizWebrtcEndpoint before playing audio. The ACK would be sent only after receiving a successful response from the Accept API, followed by an artificial delay. This approach ensures that the WebRTC connection is established before audio playback begins.

Connection State-Based Delay: Direct the SIP UA to wait until the WebRTC connection state is 'connected' before playing audio. This method relies on the WebRTC connection being fully established before audio playback starts.**Buffered Media Packets:** Buffer SIP media packets and send them only after the WebRTC connection is established. This approach ensures that no audio packets are lost due to premature playback.**Silence Insertion:** Insert a brief period of silence into the IVR audio before the actual audio content. This method allows the WebRTC connection to establish itself while the IVR is playing silence, reducing the likelihood of audio clipping.**Pre-accept** : According to the [RFC recommendation](#), a WebRTC agent, such as BizWebrtcEndpoint, should not transmit media until the ICE process is nearly complete. In our context, this means adopting an 'optimistic accept' strategy by invoking the [pre-accept API call](#) even before sending the SIP INVITE. According to SIP RFCs, a User Agent Client (UAC) should be prepared to receive media immediately after sending the INVITE. Therefore, it's advisable to initiate the WebRTC connection setup process prior to sending the SIP INVITE.

These solutions may have varying degrees of complexity and impact on your system. It's essential to evaluate their feasibility and potential trade-offs in the context of your specific use case.

Support

For support concerning WhatsApp Business Calling API, choose the **WaBiz: Calling API** topic when opening a [Direct Support](#) ticket.

Revision #4

Created 2026-04-01 14:44:22 UTC by New Admin

Updated 2026-04-06 17:50:53 UTC by New Admin