

VOICE / BROWSER INTEGRATION

One click. Your browser is the phone.

Embed calling in your existing CRM with backend-managed extensions, temporary browser sessions and no second agent Answer click.

Voice API 2.3.0

16 October 2026

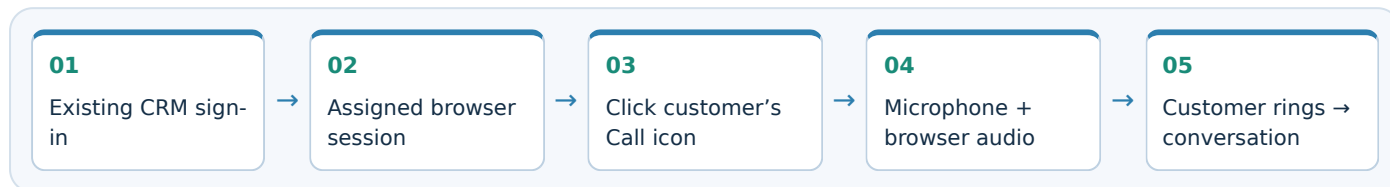
Outbound browser calling

From CRM login to browser audio

1. The experience you are building
2. Architecture and trust boundaries
3. Owner setup and CRM prerequisites
4. Backend API · Create a browser session
5. Browser SDK · Background registration, one-click dialing
6. Backend API · Create and observe the browser call
7. Session status, hangup and offboarding
8. CRM backend adapter and durable data
9. Recordings and existing call history
10. Errors, network requirements and acceptance

The experience you are building

The user signs into your existing CRM, clicks the phone icon beside a customer, and speaks through their browser. No separate portal login, extension form or agent Answer click is required.



The browser is the phone. Your CRM backend owns authentication and agent-to-extension mapping. Our service owns browser-session provisioning, gateway authorization, private PBX connectivity, existing outbound routes and call/recording APIs.

What remains unchanged

Hardphones and desktop softphones keep their existing registrations. Incoming calls continue to follow existing phone routes; this release does not migrate inbound routing to browsers. Outbound browser calls use a separate temporary device mapped to an authorized logical extension.

The existing agent-first API remains available for physical phones. Browser calls add a session ID and a per-click intent. Internally the browser leg is still established first, but the SDK accepts only its matching authorized leg automatically.

Architecture and trust boundaries

The permanent API token stays on the CRM server. The browser receives temporary, origin-bound phone configuration.



CONNECTION	ADDRESS / BEHAVIOR	AUTHENTICATION
CRM backend API	<code>https://voice-api.tychedevs.com/api/v1</code>	Organization-scoped bearer token; calls:read + calls:write.
Browser signaling	<code>wss://webphone.tychedevs.com/sip</code> with returned session/ticket query	Temporary session ticket, exact Origin and temporary SIP credentials. Use the URL returned by the API.
Browser media	Encrypted WebRTC audio to the gateway; TURN relay fallback	Per-session ICE/TURN credentials from the session response.

CONNECTION	ADDRESS / BEHAVIOR	AUTHENTICATION
Gateway → PBX	Private network connection managed by service owner	Developers need no PBX, SSH, SIP trunk or private-network credentials.

Configuration is not audio

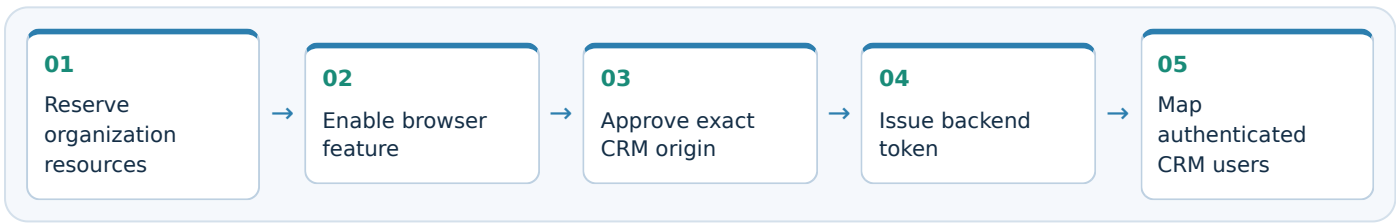
The REST API provides configuration and call control. Browser JavaScript must establish WebRTC media and obtain microphone permission. These browser credentials are inspectable by that browser user, so they are temporary and restricted; they are never the existing hardphone password.

The public browser media service is isolated from the existing PBX and uses a dedicated private trunk. Our API's call and recording identifiers remain the correlation keys.

03 / BROWSER WEBPHONE

Owner setup and CRM prerequisites

Complete these items before the developer opens a browser session.



1. In the owner console's Organizations page, reserve the client's extensions and recording locations.
2. Enable **API integration, Call activity, Outbound calling, Browser webphone integration** and the permitted mobile/landline/international categories. Enable recording metadata/audio only where needed.
3. Enter each exact HTTPS CRM origin, such as `https://crm.example.com` . No path, trailing slash or wildcard. Scheme, hostname and nondefault port matter.
4. Issue an organization-scoped token with `calls:read` and `calls:write` , plus `recordings:read` if needed. Keep it in the CRM backend's secret store.
5. The CRM backend must authenticate its own user, authorize the customer, and resolve that user's extension from a trusted database mapping. Never trust an extension or organization ID supplied by the browser.

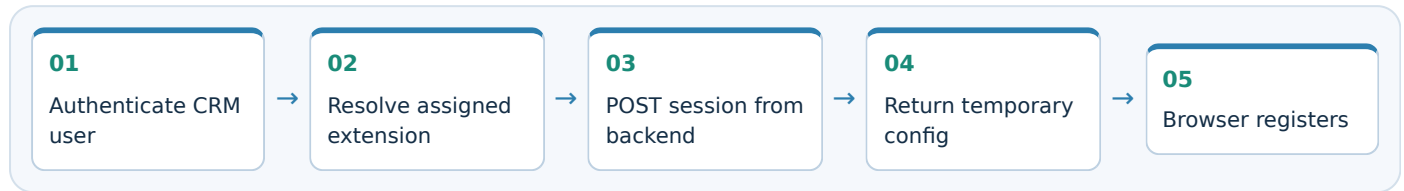
One browser session per extension

A second session returns 409 WEBPHONE_IN_USE. Do not silently steal another tab/user's session. On deliberate logout/device switch, revoke the existing session. Hardphone registration is independent and stays intact.

Your CRM must use HTTPS and permit the signaling host in its CSP, for example `connect-src 'self' wss://webphone.tychedevs.com` . Self-host the provided SDK as a CRM asset. For iframes, the parent must explicitly permit microphone access; a normal top-level CRM page is simpler.

Backend API · Create a browser session

The CRM backend supplies the authorized agent reference and extension. It returns the short-lived result only to that logged-in CRM user.



CURL · CRM BACKEND ONLY

```
curl --fail-with-body --silent --show-error \
  'https://voice-api.tychedevs.com/api/v1/webphone/sessions' \
  -H "Authorization: Bearer $VOICE_API_TOKEN" \
  -H 'Content-Type: application/json' \
  --data '{"agent_id":"crm_user_42","agent_extension":"115",
        "origin":"https://crm.example.com","ttl_seconds":3600}'
```

ILLUSTRATIVE RESPONSE · VALUES ARE PLACEHOLDERS

```
HTTP/1.1 201 Created
Cache-Control: private, no-store

{
  "id":"ws_0123456789abcdef0123456789abcdef",
  "agent_id":"crm_user_42", "extension":"115",
  "origin":"https://crm.example.com", "expires":1792155600,
  "websocket_url":"wss://webphone.tychedevs.com/sip?session=ws_...&ticket=TEMPORARY_TICKET",
  "sip_uri":"sip:ws_0123456789abcdef0123456789abcdef@webphone.tychedevs.com",
  "authorization_username":"ws_0123456789abcdef0123456789abcdef",
  "authorization_password":"TEMPORARY_BROWSER_PASSWORD",
  "ice_servers":[{"url":"turn:webphone.tychedevs.com:3480?transport=udp",
    "turn:webphone.tychedevs.com:3480?transport=tcp",
    "turns:webphone.tychedevs.com:5350?transport=tcp"}],
  "username":"EXPIRY:SESSION_ID", "credential":"TEMPORARY_TURN_CREDENTIAL"
}],
  "auto_answer":"matching_outbound_intent_only"
}
```

`ttl_seconds` is an integer from 300 to 3600 (default 3600). Expiry is capped by the integration token's expiry. `expires` is Unix seconds UTC. Store the session ID against the CRM user; keep returned secrets only as long as needed, and never log or publicly cache this response.

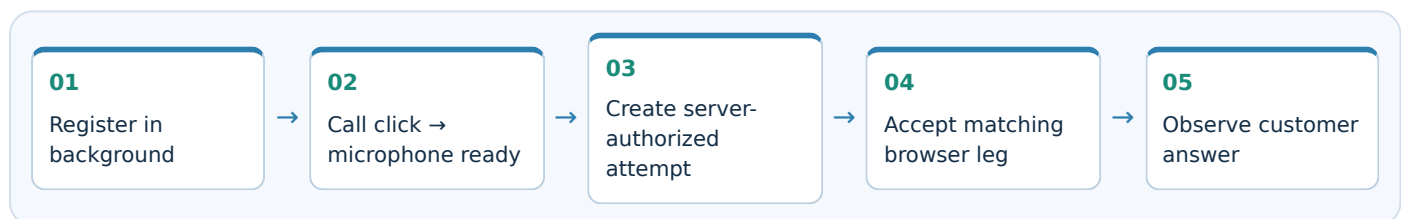
Expiry and session creation retries

The session-creation endpoint is not idempotent and does not return a lost password again. Persist the successful response securely if your CRM needs to deliver it reliably. If creation succeeded but the response was lost, use `GET /api/v1/webphone/sessions` to find your active session metadata, inspect its busy state, and explicitly revoke an idle stale session before trying again. Never revoke an active call silently or retry creation in a loop. Session expiry/revocation ends its browser channels. Provision a fresh session while idle before the next call.

05 / BROWSER WEBPHONE

Browser SDK · Background registration, one-click dialing

Import the bundled SDK into the CRM frontend. The user sees the Call button and a compact status/Mute/Hang up panel.



YOUR CRM HTML

```
<meta name="csrf-token" content="YOUR_CRM_CSRF_TOKEN">
<button id="call-customer">Call</button>
<button id="mute">Mute</button>
<button id="hangup">Hang up</button>
<p id="call-status" role="status"></p>
<audio id="remote-audio" autoplay></audio>
<script type="module" src="/assets/voice-phone.js"></script>
```

JAVASCRIPT · CRM FRONTEND

```
import {BrowserWebphone} from '/assets/browser-webphone-sdk.js';

// These /crm/voice routes are YOUR CRM routes, implemented by its developer.
// Use your existing session authentication, authorization and CSRF protection.
async function crm(path, method='GET', body) {
  const r=await fetch('/crm/voice'+path, {
    method, credentials:'same-origin',
    headers:{'Content-Type':'application/json',
      'X-CSRF-Token':document.querySelector('meta[name=csrf-token]').content},
    body:body ? JSON.stringify(body) : undefined
  });
  const data=await r.json();
  if(!r.ok)throw Error(data.error?.message || 'Phone request failed');
  return data;
}

const status=document.querySelector('#call-status');
const phone=new BrowserWebphone({
```

```

audio=document.querySelector('#remote-audio'),
createCall:body=>crm('/calls','POST',{
  clientId:body.client_id,
  browserSessionId:body.webphone_session_id,
  browserIntent:body.browser_intent
}),
readCall:id=>crm('/calls/'+encodeURIComponent(id)),
revokeSession:id=>crm('/browser-sessions/'+encodeURIComponent(id),'DELETE'),
onState:(state,call)>{
  status.textContent=state==='answered' ? 'In call' : state;
  // Start the conversation timer from call.answered_at, not browser connection.
}
});
const config=await crm('/browser-sessions','POST',{});
await phone.connect(config); // Registration is automatic; no extension login form.

const button=document.querySelector('#call-customer');
button.onclick=async()=>{
  button.disabled=true;
  try {
    // Your backend loads and authorizes the client's stored phone number.
    await phone.call({client_id:'client_123'});
  } catch(e) {
    status.textContent=e.message;
    // If submission is uncertain, resolve the stored attempt; NEVER auto-redial.
    // Enable another call only after readiness/outcome has been resolved.
  }
};
document.querySelector('#mute').onclick=()=>phone.mute(true);
document.querySelector('#hangup').onclick=()=>phone.hangup();
// On deliberate logout: revoke through CRM backend, then await phone.disconnect().

```

First-use microphone permission cannot be bypassed. A denied microphone request stops before customer origination. Enable browser audio playback through the user's interaction and expose a clear "Enable audio" action if the browser blocks playback. Re-enable the Call button after a known ended call/readiness check; keep it disabled while an attempt is active or uncertain.

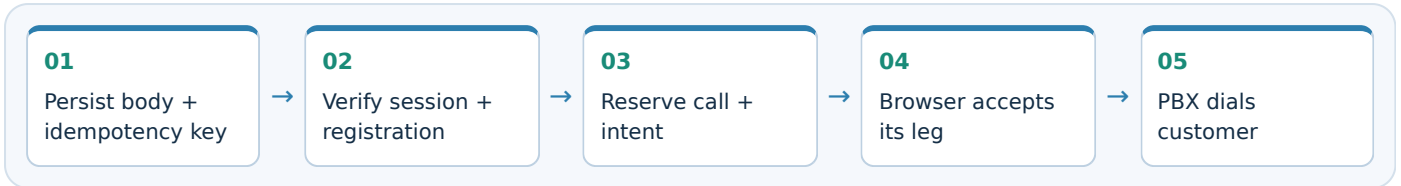
Automatic answering is tightly limited

The SDK accepts an invitation only when its server-provided X-Voice-Intent matches the random intent for this Call click, the call ID matches, and the acceptance window is still valid. Other invitations are rejected in this outbound-only SDK. It does not globally enable auto-answer, and it does not register over the hardphone identity.

06 / BROWSER WEBPHONE

Backend API · Create and observe the browser call

Keep the existing call endpoint and add the browser session and intent. Use exactly the same idempotency rules as physical-phone calls.



CURL · REAL CALL WHEN AUTHORIZED AND ENABLED

```
curl --fail-with-body --silent --show-error \
  'https://voice-api.tychedevs.com/api/v1/calls' \
  -H "Authorization: Bearer $VOICE_API_TOKEN" \
  -H 'Content-Type: application/json' \
  -H 'Idempotency-Key: PERSISTED_UUID_FOR_THIS_CRM_ATTEMPT' \
  --data '{"client_id":"client_123","agent_extension":"115",
        "destination":"09171234567","number_type":"ph_mobile",
        "webphone_session_id":"ws_0123456789abcdef0123456789abcdef",
        "browser_intent":"abcdef0123456789abcdef0123456789"}'
```

HTTP 202 returns the same existing Call schema and a call_id identifier.
Observe it using GET /api/v1/calls/{call_id} with calls:read.
Omit BOTH browser fields for existing physical-phone behavior.

`browser_intent` is 16 cryptographically random bytes expressed as 32 lowercase hexadecimal characters. The SDK generates it before the request and keeps it in browser memory for matching the incoming agent leg. The CRM backend persists it with the exact request body and idempotency key before submitting the call.

No second Answer click

The gateway verifies the session, extension and pending call intent before ringing the browser. The SDK automatically accepts only that match. Only after the browser leg answers does the PBX run the existing customer route. The customer then rings normally; “one click” does not mean the customer is instantly connected.

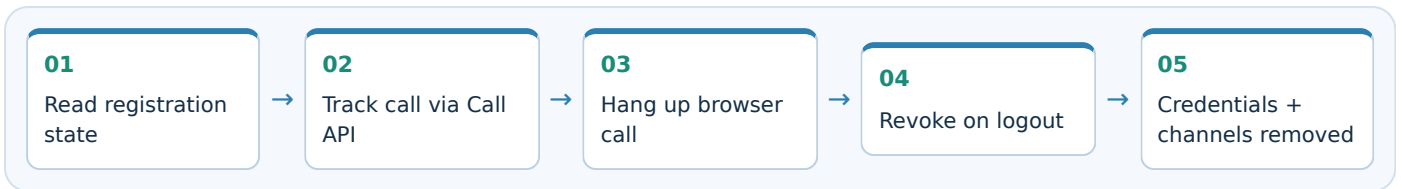
202 means accepted. Browser SIP establishment means the agent audio leg is ready; it is not evidence the customer answered. Use API `status=answered` and `answered_at` for “In call” and duration. Existing outcomes, recording delay and matching limitations remain applicable.

NUMBER TYPE	DESTINATION STRING	RULE
ph_mobile	09171234567 or +639171234567	Philippine mobile; do not add route prefixes.
ph_landline	0281234567	Leading zero and complete area code.
international	+442079460123	International +country format; owner must permit the category.

07 / BROWSER WEBPHONE

Session status, hangup and offboarding

Use the CRM backend to monitor and revoke sessions. Mute and normal Hang up operate on the browser’s SIP/media session.



CURL · BACKEND SESSION MANAGEMENT

```
# CRM backend: read session registration / busy state.
curl --fail-with-body --silent --show-error \
  'https://voice-api.tychedevs.com/api/v1/webphone/sessions/ws_0123456789abcdef0123456789abcdef' \
  -H "Authorization: Bearer $VOICE_API_TOKEN"

# Response: id, agent_id, extension, origin, expires,
#           registered:true|false, busy:true|false, status:"valid"

# CRM backend: explicit logout / cancel pending browser leg / revoke device.
curl --fail-with-body --silent --show-error -X DELETE \
  'https://voice-api.tychedevs.com/api/v1/webphone/sessions/ws_0123456789abcdef0123456789abcdef' \
  -H "Authorization: Bearer $VOICE_API_TOKEN"

# HTTP 200: {"id":"ws_...", "status":"revoked"}
```

Deletion invalidates the session in the API and removes gateway credentials/channels. A failed gateway response does not undo API revocation; gateway cleanup retries through its background policy sweep. Credential expiry, token revocation, owner/organization suspension, removed origin or disabled browser feature are rechecked about every 10 seconds. New handshakes and call starts also recheck authorization.

Do not silently drop active calls

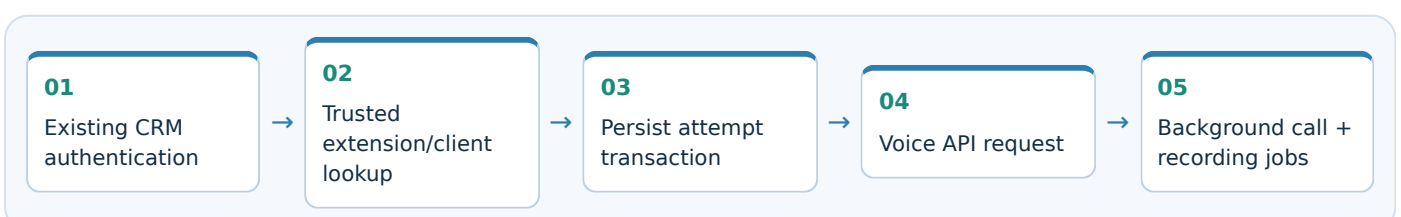
Explicit session revocation and expiry end browser calls. Warn users before logout or extension/device switching. Reconnect/renew only while idle. A lost network or closed tab can terminate audio; the CRM backend must keep tracking the existing call ID and must not auto-redial. Turning off the global outbound switch blocks new calls and does not intentionally terminate an already-established call.

Mute only changes local microphone transmission and does not end the call. A pending call without an arrived browser invitation is cancelled by revoking its browser session; reconnect a fresh session before placing a different call.

08 / BROWSER WEBPHONE

CRM backend adapter and durable data

The developer supplies a small adapter to the existing CRM database. Our downloadable backend module contains the Voice API requests; the CRM supplies its own user/customer ownership checks.



Download [voice-crm-backend.mjs](#). Import `createVoiceIntegration` and provide your server-side token, configured CRM origin and repository adapter. Do not make the example's repository methods no-ops.

REPOSITORY HOOK	REQUIRED IMPLEMENTATION
<code>authorizeVoiceAgent(user)</code>	Authorize this authenticated CRM user and return their stable agent ID and assigned extension.
<code>authorizeClientForCalling(user, clientId)</code>	Verify CRM organization/customer ownership and dialing permission. Return stored normalized phone and numberType.
<code>saveBrowserSession / authorizeBrowserSession</code>	Persist ownership by CRM user/organization; enforce that session lookups belong to the caller.
<code>reserveCallAttempt</code>	Transactional unique key on CRM user + session + browser intent. Save exact body/key before fetch; return existing attempt on retry.
<code>attachCallId / authorizeCall</code>	Persist the API call ID and enforce CRM user/customer ownership before returning call status/audio.
<code>markAttemptUncertain / closeBrowserSession</code>	Preserve uncertain attempts and revoke/close sessions on explicit logout. Never create a replacement call automatically.

CRM ROUTE WIRING

```
// Your CRM backend routes, behind existing login + authorization + CSRF middleware.
POST  /crm/voice/browser-sessions    -> integration.openSession(crmUser)
GET    /crm/voice/browser-sessions/:id -> integration.sessionState(crmUser, id)
DELETE /crm/voice/browser-sessions/:id -> integration.closeSession(crmUser, id)
POST   /crm/voice/calls              -> integration.call(crmUser, body)
GET    /crm/voice/calls/:id          -> integration.getCall(crmUser, id)

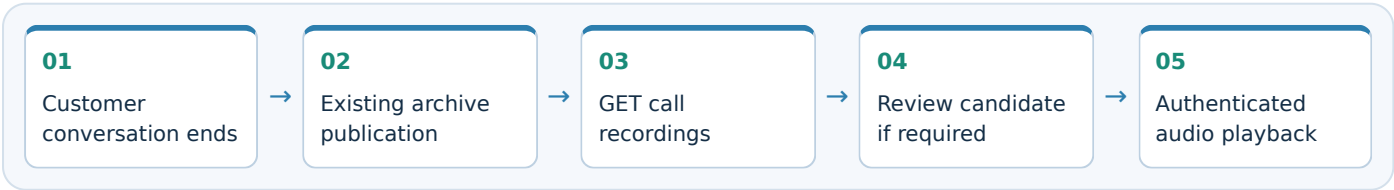
// These are suggested CRM routes, NOT additional endpoints on our Voice API.
// A per-call durable job should poll outcomes/recording links after UI disconnects.
```

For multiple agents, poll authorized call changes centrally into your CRM database and serve browser status from that cache. The API allows 240 requests/minute per principal; one upstream poller per browser will exhaust that limit as agent counts grow. Serialize attempt creation with a database transaction/unique constraint. If a POST response is lost, recover and replay the exact saved body/key using the same integration credential. Never generate a new browser intent or call key as an automatic network retry.

09 / BROWSER WEBPHONE

Recordings and existing call history

Browser audio follows the PBX conversation. Use the same call ID and recording API documented in the main integration playbook.



HTTP

```
GET /api/v1/calls/{call_id}
GET /api/v1/calls/{call_id}/recordings
GET /api/v1/recordings?date=2026-10-16&agent_extension=115
GET /api/v1/recordings/{recording_id}
GET /api/v1/recordings/{recording_id}/audio

# Include Authorization: Bearer YOUR_BACKEND_TOKEN on every API request.
# Use recordings:read, plus calls:read for call-bound recording lookup.
# Audio also requires the organization's recording-audio feature.
# Candidate confirmation remains a separately authorized reconciler operation.
```

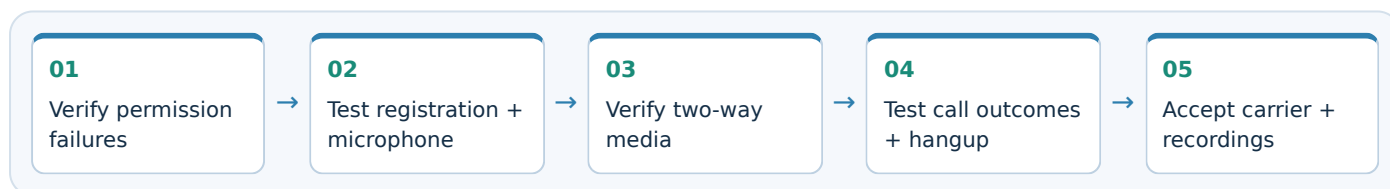
Keep the authorized logical extension as recording metadata; the temporary browser identity is not the agent's business identity. The source remains the existing PBX archive and publication workflow. Delayed availability, manual association and `completeness=unverified` remain possible. Browser integration does not repair recording publication issues.

See [CRM Call & Recording Integration Playbook](#) for the detailed recording lifecycle and authenticated audio proxy examples.

10 / BROWSER WEBPHONE

Errors, network requirements and acceptance

Handle readiness and uncertainty explicitly. One successful registration alone is not proof of usable audio.



RESULT	MEANING / ACTION
401 / 403	Token/owner/org inactive, missing scope/feature or CRM origin not approved. Correct configuration; do not fall back to an owner-wide token.
409 WEBPHONE_IN_USE	Another active browser session uses this extension. Explicitly close/revoke it; do not steal registrations.
409 WEBPHONE_NOT_READY	No usable registration or device busy. Prepare/register the browser before creating a call.
409 WEBPHONE_BUSY	Active/uncertain call or reused browser intent. Resolve the original attempt; no automatic redial.
410 WEBPHONE_EXPIRED	Session expired/revoked. Establish a new authorized session while idle.
429	Session capacity, organization quota or rate limit. Back off; honor Retry-After where supplied.
503 WEBPHONE_GATEWAY_UNAVAILABLE	Gateway dependency unavailable. Preserve exact call attempt/key; inspect returned call state on retry.

RESULT	MEANING / ACTION
Call status unknown / network timeout	Outcome uncertain. Keep the call ID/key/body and request review. Do not start another call automatically.

The CRM page and WSS signaling use TCP443. Direct browser media uses gateway UDP30000–30199. TURN is available on UDP/TCP3480 and TLS5350, with gateway relay UDP50000–50199. Restrictive corporate networks must permit an appropriate path; this deployment does not promise TURN-over-TLS on TCP443. The existing website’s HTTPS/Tunnel protection does not carry WebRTC audio.

1. Verify first-use microphone permission denied/allowed, headset audio and playback permission.
2. Verify browser registration, origin restrictions and that an unrelated invitation is never auto-answered.
3. Place a controlled call to a destination you own: no agent Answer click, customer rings, two-way audio, correct outcome.
4. Verify Mute and Hang up from either party; pending-call cancellation; browser closure and network failure.
5. Test approved mobile/landline/international routes and recording metadata/audio under the same logical extension.
6. Test token/session revocation, expiry, feature/origin changes, duplicate clicks and tenant isolation.
7. Test the actual client networks, including a forced TURN relay test and corporate firewall conditions.

Verification status

Verification: 72 backend tests, 15 portal tests, matching-intent SDK tests, live WSS registration, wrong-origin/ticket rejection, raw SIP dialing rejection, server-side session revocation, and private PBX echo with actual two-way RTP packets. A forced TURN relay path was also verified. Browser Mute/Unmute and Hang up were checked against the sender and PBX call leg. No external customer/carrier test is performed without an approved destination. Carrier routing and full recording coverage remain controlled acceptance checks. The reference video demonstrates UI behavior, not proof of our deployed carrier audio.

[OpenAPI 2.3.0](#) · [Existing quickstart](#) · [Owner console browser integration section](#)

Browser Webphone Integration · Voice API 2.3.0 · Credentials delivered separately · Examples use placeholders.