

# WhatsApp Gateway Documentation

Generated at 2026-05-13 09:12:14

## WhatsApp Gateway Documentation

Generated on: 2026-05-13 09:12:14

### Environment

- Base URL: 'https://evoapi.faisak.com'
- API Authorization: 'Authorization: Bearer YOUR\_API\_KEY'
- Connection: 'WhatsApp Gateway managed connection'

### Quickstart (Step by Step)

1. Test server reachability:

```
curl -i "https://evoapi.faisak.com/"
```

2. Create an API key in the customer portal:

```
Dashboard > API Keys > Create API Key
```

3. Create instance in app dashboard only:

```
Dashboard > Instances > Create Instance
```

4. Connect and monitor the instance from the customer portal.

### Responsible Messaging

WhatsApp Gateway is designed for transactional, operational, support, authentication, and internal team messages. Marketing is not allowed.

Allowed 'message\_purpose' values:

- 'transactional'
- 'operational'
- 'support'
- 'internal\_team'
- 'authentication'

Disallowed values:

- 'marketing'
- 'promotion'
- 'cold\_outreach'
- 'bulk\_campaign'
- 'political'

### Send Messages

# WhatsApp Gateway Documentation

Generated at 2026-05-13 09:12:14

WhatsApp Gateway is for transactional, operational, and customer lifecycle messages. Marketing is not allowed. Do not use it for cold outreach, scraped lists, spam, or political messaging.

Every send payload should include 'message\_purpose'. Allowed values are 'transactional', 'operational', 'support', 'internal\_team', and 'authentication'. Marketing, promotion, cold outreach, bulk campaign, political, and unknown purposes are not allowed.

Example transactional request:

```
POST /api/send-message
Authorization: Bearer YOUR_API_KEY
Content-Type: application/json
```

```
{
  "instance_id": "inst_12345",
  "phone": "+9665XXXXXXX",
  "message": "Your order #2041 has been shipped.",
  "message_purpose": "transactional"
}
```

Example response:

```
{
  "success": true,
  "status": "sent",
  "message_id": "string",
  "log_id": 123
}
```

The send endpoint requires 'Content-Type: application/json' and 'Authorization: Bearer CUSTOMER\_API\_KEY'. Manage API keys in the customer dashboard; '/api-keys' is not a public JSON API endpoint.

Send text through the guarded API:

```
curl -i -X POST "https://evoapi.faisak.com/api/send-message" \
-H "Content-Type: application/json" \
-H "Authorization: Bearer YOUR_API_KEY" \
-d '{"instance_id":"inst_12345","phone":"+9665XXXXXXX","message":"Your invoice INV-2041 is ready. Please check your customer portal.","message_purpose":"transactional"}'
```

Media sending is not exposed through the public beta API yet.

## Receive Messages (Webhooks)

WhatsApp Gateway manages the internal connection to the messaging provider automatically. Customers only configure the events they want WhatsApp Gateway to forward to their own systems.

1. WhatsApp Gateway receives messages from the connected WhatsApp number.

# WhatsApp Gateway Documentation

Generated at 2026-05-13 09:12:14

2. Messages appear in Inbound Logs.
3. Optional customer webhooks can forward selected events to CRM, ERP, n8n, Make, 3CX, or custom apps.
4. Customers select normalized event types such as 'message.received' or 'recipient.opted\_out'.

Customer webhook selection example:

```
{ "destination_url": "https://your-app.example.com/webhooks/whatsapp-gateway", "events": [ "message.received", "recipient.opted_out", "message.status_updated" ] }
```

Opt-out webhook event example:

```
{ "event": "recipient.opted_out", "instance_id": "demo-support-line", "instance_name": "Demo Support Line", "from": "+9665XXXXXXX", "reason": "recipient_requested_stop", "message_text": "STOP", "timestamp": "2026-05-08T10:00:00Z" }
```

## Plan Limits

Each plan includes daily and monthly fair-use limits. Limits protect gateway reliability and prevent misuse.

Plan	Daily limit	Monthly fair use
Starter	100/day	2,000/month
Growth	500/day	10,000/month
Scale	1,500/day	30,000/month

## Responsible Messaging Error Responses

Responsible messaging and limit failures use this JSON shape:

```
{
  "success": false,
  "error": "error_code",
  "message": "Human readable explanation.",
  "details": {
    "action": "block",
    "risk_score": 90,
    "limit": 100,
    "used": 100
  }
}
```

Error	HTTP	Message
'unsupported_media_type'	415	This endpoint requires application/json.
'invalid_api_key'	401	A valid API key is required.

# WhatsApp Gateway Documentation

Generated at 2026-05-13 09:12:14

`revoked\_api\_key`	401	This API key has been revoked.
`invalid\_json`	400	Request body must be valid JSON.
`validation\_error`	422	The request payload is missing required fields.
`module\_not\_enabled`	403	API access is not enabled for this account.
`active\_plan\_required`	403	An active plan is required before sending messages.
`invalid\_instance`	404	The requested instance was not found for this API key.
`message\_purpose\_required`	422	message\_purpose is required. Allowed values are transactional, operational, support, internal\_team, and authentication.
`message\_purpose\_not\_allowed`	422	WhatsApp Gateway does not support marketing, cold outreach, political, or bulk campaign messaging.
`daily\_message\_limit\_exceeded`	429	Your daily transactional message limit has been reached. Please try again tomorrow or request a higher reviewed limit.
`monthly\_message\_limit\_exceeded`	429	Your monthly fair-use message limit has been reached. Please contact support for a reviewed higher limit.
`recipient\_suppressed`	403	This recipient has opted out or has been blocked from non-essential messages.
`sending\_throttled`	429	Sending speed has been temporarily limited to protect gateway reliability and prevent bulk messaging.
`instance\_paused\_for\_review`	423	This instance has been paused for responsible messaging review
`policy\_violation`	403	This message appears to violate WhatsApp Gateway responsible messaging rules. Bulk marketing, spam, and cold outreach are not allowed.
`gateway\_send\_failed`	502	The message passed responsible messaging checks, but delivery could not be completed. Please try again or contact support.
`internal\_error`	500	The request could not be processed. Please try again or contact support.

## Example limit error:

```
{
  "success": false,
  "error": "daily_message_limit_exceeded",
  "message": "Your daily transactional message limit has been reached. Please try again tomorrow or request a higher reviewed limit.",
  "details": {
    "action": "block",
    "risk_score": 90,
    "limit": 100,
    "used": 100
  }
}
```

## Best Practices

- Send messages only to expected recipients.
- Keep messages transactional and relevant.
- Avoid repeated identical content.
- Avoid suspicious or shortened links.

# WhatsApp Gateway Documentation

Generated at 2026-05-13 09:12:14

---

- Respect opt-outs.
- Monitor failed delivery rates.
- Use official Meta/BSP channels where approved marketing is required.

## 3CX Middleware Overview

3CX middleware connects inbound support workflows to controlled transactional WhatsApp replies through WhatsApp Gateway.

What the integration does:

- Forward inbound WhatsApp messages into 3CX workflows.
- Reply to active WhatsApp conversations from 3CX.
- Send controlled missed-call, voicemail, call-ended, queue callback, and after-hours replies only inside the active conversation window.
- Keep fair-use limits, opt-outs, pause controls, and policy checks active.

Setup requirements:

- Customer account.
- At least one WhatsApp instance.
- 3CX integration settings saved in the customer dashboard.
- Webhook secret copied into 3CX.

Responsible messaging note: 3CX middleware is not a marketing, campaign, broadcast, bulk messaging, or cold outreach sender. The WhatsApp user must message first, and 3CX replies are limited to the active conversation window. Outbound messages pass through responsible messaging controls, including message purpose validation, fair-use limits, suppression checks, pause controls, and policy logging.

Security options:

- Webhook secret validation.
- Optional IP allowlist.
- Optional HMAC signing.
- 256KB payload limit.
- Rate limiting.
- Sanitized logs.
- Admin monitoring.

## 3CX Webhook Reference

Endpoint:

```
POST /integrations/3cx/webhook/{integration_id}
X-WhatsApp-Gateway-3CX-Secret: *****
Content-Type: application/json
```

# WhatsApp Gateway Documentation

Generated at 2026-05-13 09:12:14

---

Optional HMAC headers:

- 'X-WhatsApp-Gateway-Timestamp'
- 'X-WhatsApp-Gateway-Signature'

Preferred signature format: 'X-WhatsApp-Gateway-Signature: sha256=<hex>', where '<hex>' is HMAC SHA256 of 'timestamp + "." + raw\_request\_body'. Bare '<hex>' signatures are also accepted for compatibility.

Before a 3CX webhook request is authenticated, WhatsApp Gateway returns a generic response for unknown or disabled integrations, missing or wrong secrets, IP allowlist failures, stale timestamps, and invalid signatures:

```
{
  "success": false,
  "error": "invalid_3cx_webhook",
  "message": "Invalid 3CX webhook request."
}
```

After authentication, safe specific errors may include 'payload\_too\_large', 'webhook\_rate\_limited', 'invalid\_json', 'module\_not\_enabled', 'missing\_phone', 'missing\_message\_text', 'message\_purpose\_not\_allowed', 'recipient\_suppressed', 'sending\_throttled', 'policy\_violation', or 'gateway\_send\_failed'.

Supported 3CX event types:

- 'test'
- 'outbound\_message'
- 'inbound\_reply'
- 'missed\_call'
- 'call\_started'
- 'call\_ended'
- 'voicemail'
- 'queue\_event'
- 'queue\_callback'
- 'after\_hours'

Outbound reply example:

The 'outbound\_message' event is a 3CX reply into an active WhatsApp conversation. The WhatsApp user must have messaged first, and the last inbound message must be within the active conversation window.

```
{
  "event_type": "outbound_message",
  "to": "+966500000000",
  "message_text": "Your support request has been updated.",
  "message_purpose": "support",
}
```

# WhatsApp Gateway Documentation

Generated at 2026-05-13 09:12:14

---

```
"conversation_id": "3cx-chat-123",  
"agent_extension": "101"  
}
```

Allowed 3CX 'message\_purpose' values:

- 'support'
- 'operational'
- 'transactional'
- 'internal\_team'
- 'authentication'

Blocked purpose values:

- 'marketing'
- 'promotion'
- 'cold\_outreach'
- 'bulk\_campaign'
- 'political'
- 'unknown'

## 3CX Call Event Templates

Call events can send controlled WhatsApp replies only when 'call\_event\_enabled' is active, the relevant template is enabled, and strict conversation-window mode allows the recipient.

Supported call events:

- 'missed\_call'
- 'voicemail'
- 'call\_ended'
- 'queue\_callback'
- 'after\_hours'

Template variables:

- '{{customer\_phone}}'
- '{{agent\_extension}}'
- '{{queue}}'
- '{{call\_time}}'
- '{{business\_name}}'

Call-event safety behavior:

- 15-minute cooldown per integration, event type, and destination phone.
- Strict conversation-window mode is enabled by default, so the WhatsApp user must have messaged first within the active window.

# WhatsApp Gateway Documentation

Generated at 2026-05-13 09:12:14

---

- Suppressed recipients are blocked.
- All messages pass through 'ResponsibleMessageSender'.
- Marketing language and disallowed purposes are blocked.

## 3CX Middleware Troubleshooting

| Error | Cause | Fix |

| 'invalid\_3cx\_webhook' | The request could not be authenticated before processing. | Check integration URL, webhook secret, optional IP allowlist, timestamp, and HMAC signature. |

| 'payload\_too\_large' | Webhook body exceeds 256KB. | Reduce payload. |

| 'webhook\_rate\_limited' | More than 60 requests per minute per integration. | Slow down the event sender. |

| 'invalid\_json' | Request body is not valid JSON. | Send a valid JSON payload with Content-Type: application/json. |

| 'module\_not\_enabled' | 3CX Middleware is not enabled for the account. | Enable 3CX in module settings or contact support. |

| 'conversation\_window\_closed' | The recipient has not messaged first or the last inbound WhatsApp message is outside the active conversation window. | Ask the customer to message the WhatsApp number first, then reply from 3CX within the active window. |

| 'outbound\_disabled' | Outbound messaging disabled in integration settings. | Enable outbound. |

| 'call\_events\_disabled' | Call events disabled. | Enable call events. |

| 'missing\_phone' | Event payload does not include customer phone. | Include to, from, customer\_phone caller, or phone. |

| 'missing\_message\_text' | Outbound event does not include message text. | Include message\_text or message in the event payload. |

| 'message\_purpose\_not\_allowed' | The message purpose is missing or not allowed. | Use support, operational, transactional, internal\_team, or authentication. |

| 'recipient\_suppressed' | Recipient opted out or was blocked. | Review suppressed recipients. |

| 'sending\_throttled' | Sending speed was temporarily limited. | Retry later and review sending volume. |

| 'gateway\_send\_failed' | Delivery could not be completed after responsible messaging checks. | Retry later or contact support if it continues. |

| 'daily\_message\_limit\_exceeded' | Customer plan limit reached. | Request a higher reviewed limit. |

| 'policy\_violation' | Message blocked by responsible messaging policy. | Review purpose, content, recipient source, and usage pattern. |

## Core Endpoint Map

| Operation | Method | Path |

| Health / Root | 'GET' | '/' |

| Send Message | 'POST' | '/api/send-message' |

| Manage Instance | 'GET' | '/instances/{id}/manage' |

| Suppressed Recipients | 'GET' | '/suppressed-recipients' |

| Acceptable Use | 'GET' | '/acceptable-use' |

## Customer Integration Areas

- API Keys: used by customer systems to send messages through WhatsApp Gateway.

# WhatsApp Gateway Documentation

Generated at 2026-05-13 09:12:14

---

- Send Message API: `/api/send-message` for guarded support, transactional, or operational messages.
- Customer Webhooks: forward selected WhatsApp events to your CRM, ERP, n8n, Make, or custom app.
- Inbound Logs: confirm WhatsApp messages received by connected numbers.
- 3CX Integration: configure a customer destination webhook for 3CX middleware.
- Responsible Messaging: transactional, support, operational, and other controlled use only.

Area	Group	Path
API Keys	Customer Access	<code>/api-keys</code>
Send Message API	Messaging	<code>/api/send-message</code>
Customer Webhooks	Events	<code>/instances/{id}/manage?tab=webhook</code>
Inbound Logs	Visibility	<code>/inbound-logs</code>
3CX Integration	Integrations	<code>/integrations/3cx</code>
Responsible Messaging	Safety	<code>/docs/responsible-messaging</code>

## Supported Webhook Events

### ### Recommended Events

- `message.received`
- `message.sent`
- `message.failed`
- `recipient.opted_out`

### ### Instance Events

- `instance.connected`
- `instance.disconnected`
- `instance.connection_updated`
- `instance.qr_updated`

### ### Message Lifecycle Events

- `message.delivered`
- `message.read`
- `message.status_updated`

### ### Conversation Events

- `conversation.updated`

### ### Call Events

- `call.received`
- `call.missed`

### ### Group Events

- `group.message_received`
- `group.updated`

# WhatsApp Gateway Documentation

Generated at 2026-05-13 09:12:14

---

## Troubleshooting

- 401/403: verify 'Authorization: Bearer YOUR\_API\_KEY'.
- 404: check that you are calling the documented WhatsApp Gateway route for your account.
- 500 with 'SQLSTATE[HY000] [1045]': fix MySQL credentials/grants for portal DB.
- 'registration\_enabled fallback used': DB settings read failed; restore DB connectivity.
- Send works but no forwarded event: verify Customer Webhook settings and receiver HTTP 200 behavior.

## Portal Paths

- Docs Overview: '/docs'
- Quickstart: '/docs/quickstart'
- Connect: '/docs/connect'
- Send: '/docs/send'
- Receive: '/docs/receive'
- Responsible Messaging: '/docs/responsible-messaging'
- 3CX Middleware Overview: '/docs/3cx'
- 3CX Webhook Reference: '/docs/3cx/webhooks'
- 3CX Call Event Templates: '/docs/3cx/call-events'
- 3CX Middleware Troubleshooting: '/docs/3cx/troubleshooting'
- Endpoints: '/docs/endpoints'
- Troubleshooting: '/docs/troubleshooting'