Webhooks

Alchemer Voice sends a signed HTTP POST with a JSON body to an address you choose when selected events occur, so your service does not need to poll the API.

Events

Event	Sent when	Payload
`conversation_processing_complete`	Alchemer Voice has finished processing a conversation.	The conversation id. Use it to fetch the conversation, summary and insights from the OData API.
`workflow`	A Webhook step in an Alchemer Voice Workflow runs. Tick this event on your webhook; the workflow must start from a conversation, insight or timestamp trigger.	The workflow id and a list of contexts the workflow was running on, such as conversations, insights, topics, topic models, labels, an Analyst chat or a timestamp.

Create a webhook

Before you begin: Alchemer Voice Admin privileges. Creating webhooks needs the Manage connections permission, which admins have by default.

1. Build an endpoint that accepts POST requests with a JSON body.
2. In Alchemer Voice, go to **Admin** > **Connections** and scroll to **Webhooks**.
3. Click **Add webhook**, enter the Endpoint URL and tick the Events you want.
4. Optionally choose a Linked OAuth client (see Linked OAuth clients below), then click **Create**.
5. Copy the Webhook Secret from the webhook card (hidden until you click it). You need it to verify requests.
6. Click the card to change events or the endpoint, or to delete it.

Each webhook's secret is created with it. To get a new one, delete the webhook and add it again.

The endpoint must resolve to a public internet address. Localhost, private networks and other internal ranges are rejected. For local development, use a tunnelling service.

Request format

Each request is a POST with a JSON body, the user agent "Four/Four" and a header named `Signature`.

Body field	Description
`timestamp`	When the event was sent (ISO 8601).
`event`	The event name.
`tenant`	Id of the Alchemer Voice workspace that raised the event.
`payload`	Event-specific details; for `conversation_processing_complete`, an object containing the conversation id.

Verifying the signature

The `Signature` header is a hex-encoded HMAC-SHA256 of the exact JSON text of the request body, keyed with your webhook secret.

1. Read the raw request body before parsing it.
2. Compute HMAC-SHA256 of that raw body with your secret.
3. Compare it with the `Signature` header, ideally with a constant-time comparison.
4. If they differ, discard the request.

Delivery and retries

Alchemer Voice waits 3 seconds for a response. Reply with a 2xx status quickly and do the real work afterwards.

If your endpoint is slow or returns an error, Alchemer Voice retries, up to three attempts in total, with increasing delays. Make your handler safe to run twice for the same event.

Linked OAuth clients

A webhook with no linked client receives events only from the workspace it was created in. If you build a product on an OAuth client (see the API access tokens and OAuth clients article), link the webhook to that client. It then receives events from every workspace where a user has authorised your client, and the `tenant` field identifies the source workspace.

Troubleshooting

- **"Endpoint URL is not valid":** use a full https address reachable from the internet.
- **Nothing arrives:** check the event is ticked, your server responds within 3 seconds, and your firewall allows internet requests.
- **Signature mismatch:** compute over the raw body, not a re-serialised copy, using this

webhook's secret.

If you need help, contact Alchemer Support.

Related Articles