

Chatwoot connector

Use the Chatwoot connector when an AIP application needs governed access to one Chatwoot account. The connector exposes 150 fixed account API operations, three conversation-oriented AIP composites, and optional authenticated webhook ingress

SECTION	Connectors
TYPE	Connector
PROTOCOL	AIP 1.0
SOURCE	docs/connectors/chatwoot/README.md

The connector inserts its configured account ID, base URL, and API token inside the host. An Action can choose an admitted capability and provide its remaining path, query, or body values, but it cannot select another Chatwoot account or origin.

At a glance

Property	Reviewed value
Connector ID	chatwoot
Standalone process	aip-host-chatwoot
AIP workspace version	2.0.0
Reviewed AIP source	d7cce13d1d555644d04a4d73c66c95b113737635
Pinned Chatwoot source	8818d276b954ac4f84cffd8915c99f40e43804ed
Connector profile	aip.connector.chatwoot.v1
Provider capability shape	cap:chatwoot:<operation.suffix>
Full capability count	150 provider operations plus three AIP composites
Provider authentication	Deployment-owned api_access_token header
Inbound path	Optional POST /webhooks/chatwoot
Streaming and provider cancellation	Not implemented

The full 153-capability count applies only when the operation allowlist is omitted or contains all 150 provider operations. A deployment may publish an exact non-empty subset of the provider catalogue; the three composites remain present.

These numbers describe the source contract. They do not prove compatibility with an unidentified Chatwoot image, edition, configuration, or extension.

Choose this connector when

The connector fits an integration that needs to:

- manage account agents, teams, bots, inboxes, integrations, and webhooks;
- create, assign, update, search, and report on customer conversations;
- send messages and manage contacts, companies, labels, and related records;

- administer automation rules, macros, canned responses, and portal content;
- consume verified Chatwoot webhook deliveries as durable AIP events;
- restrict one deployment to an edition-specific or policy-specific operation

allowlist.

Use another integration boundary when the task needs an arbitrary Chatwoot URL or method, multiple Chatwoot accounts in one connector instance, a provider-native streaming result, or a route absent from the pinned catalogue.

Re-audit and requalify the exact provider and connector artifacts before advancing the Chatwoot source pin.

Understand the standalone account boundary

One connector instance owns one Chatwoot origin and account:

1. An operator configures the base URL, account ID, and owner-only API token file.
2. The host optionally applies an exact operation-suffix allowlist.
3. Registry admission binds the resulting manifest, artifact, tenant, external account, credential revision, instance, and replicas.
4. An authenticated caller submits an Action to product-neutral `getaip-server`.
5. The registry selects the tenant-owned Chatwoot instance and a ready replica.
6. The host replaces `{account_id}` with its configured value and encodes the remaining path segments from validated input.
7. The connector adds the API token and sends one fixed method and route.
8. The result returns through the durable AIP lifecycle.

The base URL must be an HTTPS origin without credentials, path, query, or fragment. Explicit loopback HTTP is accepted for controlled local use. Authenticated provider requests do not follow redirects.

Changing the account, origin, allowlist, upstream pin, or manifest is a deployment and admission change. Do not treat it as request-level recovery.

Choose a capability family

The 150 provider operations are organized into eight reader-facing families.

Family	Operations	Included domains
Accounts, agents, teams, and bots	22	Account, agents, assignable agents, teams, members, and agent bots
Inboxes, integrations, and webhooks	21	Inboxes, members, health, templates, integration hooks, and webhooks
Conversations and messages	28	Conversation and message lifecycle, assignment, labels, typing, and attachments
Contacts and companies	31	Contact and company lifecycle, notes, merge, import, export, and relations
Automation, macros, and canned responses	16	Rules, cloning, macro execution, and canned content
Labels, attributes, and filters	15	Labels, custom attribute definitions, and custom filters

Family	Operations	Included domains
Reports, audit, and events	12	Reports, summary reports, reporting events, and audit log
Customer portals	5	Portals, categories, and articles
Provider catalogue	150	Complete frozen operation set

The catalogue contains 62 `GET`, 47 `POST`, 21 `PATCH`, and 20 `DELETE` operations. Automation-rule clone, macro execute, and conversation transcript are workflows. The other 147 catalogue entries are tools.

The three AIP composites are separate capabilities:

Capability	Outcome
<code>cap:chatwoot:message:create</code>	Send one public reply or private note and mark it with the AIP Action identity
<code>cap:chatwoot:conversation:status</code>	Set a conversation to <code>open</code> , <code>resolved</code> , or <code>pending</code>
<code>cap:chatwoot:conversation:handoff</code>	Assign a conversation, add a private note, or perform both in sequence

The composite handoff is not one atomic Chatwoot transaction. Its focused reference page defines partial-effect and recovery decisions.

Keep credentials and account identity outside Actions

Every provider operation uses the configured account ID. Input schemas omit `account_id` even though the fixed provider templates contain that segment. Other path parameters are bounded and encoded as path segments.

The host reads the API token from an owner-only file and adds it as `api_access_token`. It does not accept a token or base URL in Action input. The HTTP client refuses cross-origin redirects, which prevents automatic token forwarding to another origin.

Use one connector instance per Chatwoot account and AIP tenant boundary. Give each instance its own API token, runtime database role, admission package, external-account identity, and webhook secret. An upstream token still needs the least Chatwoot privileges required by the admitted suffix set.

Apply mutation and retry rules

All 88 non-`GET` provider operations require human approval and an AIP idempotency key scoped to the external account. The connector forwards the Action ID and idempotency key as provider headers for catalogue operations. That header does not prove the selected Chatwoot deployment deduplicates a mutation.

Provider catalogue reads declare safe retry support. Mutations declare unsafe retry behavior and rollback is not implemented. All three composites require approval and idempotency; only the conversation-status composite declares retry support.

The connector has no provider cancellation implementation. If local cancellation wins before a provider response, the result is uncertain. Read the original Action and Chatwoot state before deciding whether any mutation can be replaced or reconciled.

Customer, contact, conversation, and message data is marked restricted, PII-bearing, and subject to redaction. Keep raw bodies, attachments, tokens, and personal data out of logs and evidence summaries.

Treat webhook ingress as a separate trust path

Webhook ingress is absent when no webhook-secret file is configured. When enabled, the standalone host:

1. limits the exact request body to 1 MiB;
2. requires delivery, timestamp, and `sha256` signature headers;

3. verifies HMAC over the timestamp, a dot, and the original UTF-8 body;
4. rejects timestamps outside a five-minute window;
5. records delivery identity in instance-scoped durable profile state;
6. ignores messages marked as produced by the Chatwoot connector;
7. derives a deterministic AIP event ID from the authenticated delivery ID;
8. appends that event durably before enqueueing central publication.

A provider retry of the same accepted delivery is projected to the same event identity. This supports recovery when a process fails after replay admission or event persistence. It does not authorize another outbound response.

An accepted HTTP response can include a queued central-delivery state. It does not prove every downstream consumer has processed the event.

Deliberate exclusions

The reviewed connector does not provide:

- arbitrary provider path, method, origin, or account selection;
- multiple Chatwoot accounts in one standalone instance;
- provider credentials from protocol messages or generic credential handles;
- operation-specific schemas for every upstream request body;
- native streaming, asynchronous completion, provider cancellation, AIP

transactions, reconciliation, or rollback;

- safe automatic replay of a Chatwoot mutation;
- a webhook route when verification or central event publication is not configured;
- compatibility with a Chatwoot revision other than the pinned source;
- production readiness for an unqualified artifact and deployment.

The generic catalogue schema bounds paths, query values, JSON bodies, and multipart data. Chatwoot remains responsible for operation-specific semantic validation.

Evidence status

Evidence	What it supports	What it does not support
Source and 150-route matrix	Implemented account-scoped contract at the reviewed AIP commit	Execution of an unbuilt artifact
Connector unit tests	Deterministic catalogue, request, webhook, replay, and redaction checks exist	A retained pass for a promoted image
Controlled product fleet	Admission of 153 capabilities and selected fixture-backed reads and mutation	Real Chatwoot compatibility
Live-product inventory	No dedicated external Chatwoot harness exists at the pinned revision	Any current isolated-live or production pass

Qualify the exact connector image, Chatwoot artifact, account configuration, webhook path, and failure cases before publishing a readiness claim.

Plan the next task

Continue according to the task you need to complete:

- use `getting-started/quickstart.md` for one admitted host and first conversation flow;
- use the authentication page to bind the tenant, account, and least-privilege token;
- use the configuration reference for host, ingress, limit, and allowlist settings;
- open `capabilities/README.md` to choose one of the eight provider families or three composites;
- use the webhook and mutation references before enabling inbound events or state changes;
- use operations and qualification pages before promoting an artifact.

The global pages below own shared fleet, source-identity, and evidence rules.

Related documentation

- [Connector documentation](#)
- [Pinned upstream baselines](#)
- [Connector fleet quickstart](#)
- [Conformance and qualification](#)
- [Live-product qualification](#)